Izdaje BIGZ

OOUR „Duga"

računari 7

Specijalno izdanje avgust/septembar '85.

časopisa „Galaksija" cena 250 dinara

naš test
štampač „kanon"

računari u izlogu
komodor 128

ekskluzivno
velikani na rasprodaji

periferijska oprema
svetlosna pera

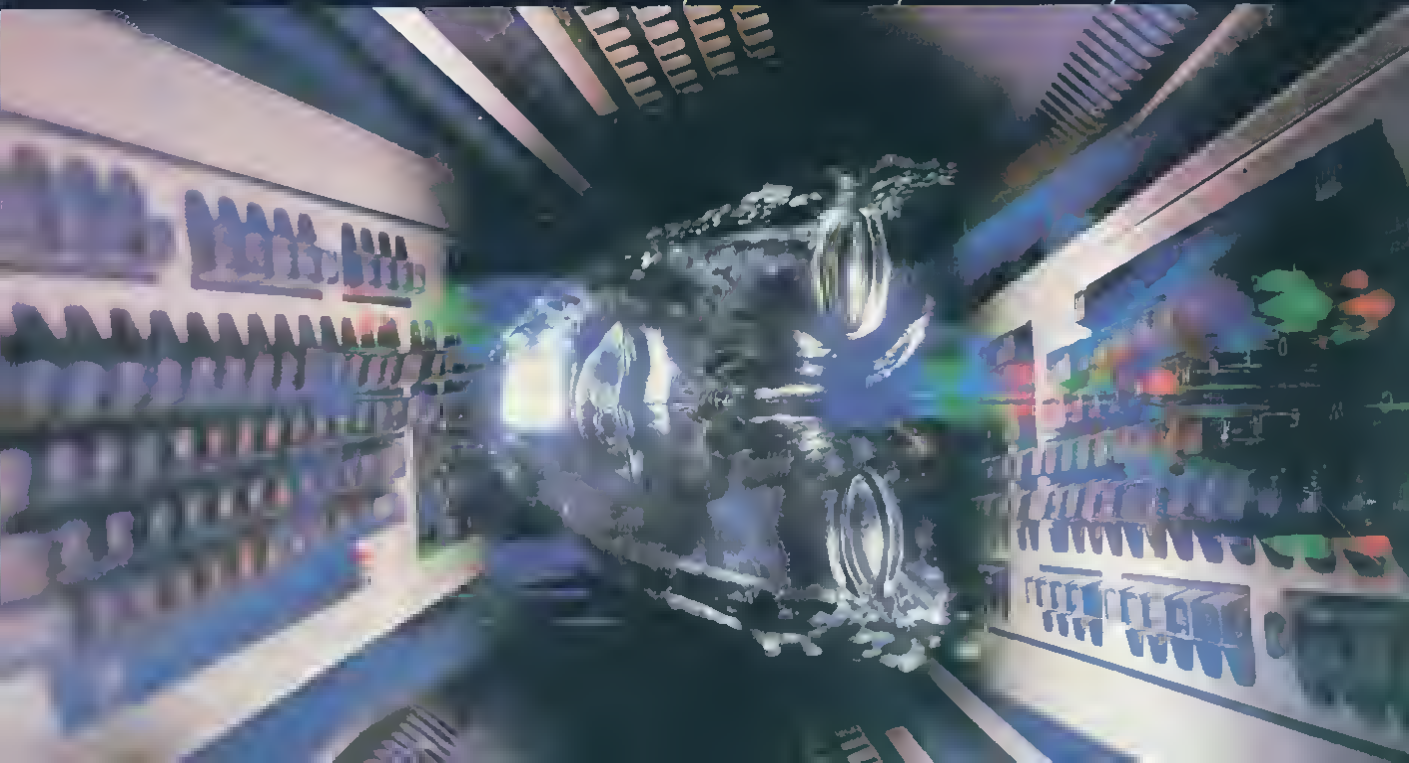
umetnost programiranja
hakeri na usijanom limenom krovu

„komodor"
periferije iz mašinca

računari u akciji
baze podataka

umetak 32 strane

SVE SPEK TRUMOVE RUTINE



računari

Specijalno izdanje
avgust septembar '85

OOUR „Duga"

časopis „Galaksija"
cena 250 dinara

Izdaje
Beogradski izdavačko-grafički zavod
OOUR Novinska delatnost „Duga"
11000 Beograd
Bulevar vojvode Mišića 17

Telefoni
650-161 (redakcija)
650-528 (prodaja)
651-793 (propaganda)

Generalni direktor
Dobrosav Petrović

Direktor OOUR „Duga"
Bratoljub Babić
Glavni i odgovorni urednik
Gavrilo Vučković
Urednik izdanja
Jova Regasek

Likovna i grafička oprema:
Raša Golubović

Redakcija časopisa „Galaksija"
Tanasije Gavranović pomoćnik glavnog i odgovornog urednika
Esad Jakupović zamenik glavnog i odgovornog urednika
Aleksandar Milinković, urednik
Jova Regašek urednik
Žarka Simović, sekretar redakcije
Špiro Stojančev, novinar
Gavrilo Vučković, glavni i odgovorni urednik

Stručna saradnja
Dejan Ristanović
Nevenka Spalević
Anđelka Zgonjanin
Mihajlo Tešanić

Autori tekstova
Vladimir Krstanović
Mihajlo Karapandžić
Zvonimir Vietrička
Branko Đaković
Vladimir Kostić
Ivan Nador dipl. ing.
Berivoj Pavlić
Bogdan Petrović
Dejan Ristanović
Jelena Rupnik
Dušan Savić
Jovan Skuljan
Zoran Živadić

Crteži
Miša Marković

Tehnička saradnja
Ljubica Milovanović
Ljev Rjaščenko

Prevodioci
Esad Jakupović
Ksanija Pjetlić-Lebedinski
Dimitrije Sedić

Izdavački savet „Galaksije"

Dr Ruđi Dekljadi, prof. dr Branislav Dimitrijević (predsednik), Radovan Dražković, Tanasije Gavranović, Živorad Gligić, Esad Jakupović, Velizar Masleš, Nikola Pajić, Željka Perunović, prof. dr Momčilo Ristić, Vlada Ristić, dr ing. Milorad Tauflović, Vidojko Veličković, Velimir Vesović, Milaje Vukelić

Štampa
Beogradsko izdavačko-grafički zavod
11000 Beograd Bulevar vojvode Mišića 17

Žiro-račun kod SDK 60802-603-2463
Devizni račun kod Beobanke
60811-620-8-82761 939-01068
Za inostranstvo cena dvostruka (400 D 2,50 US$, 6,50 DM 45 Sch, 5,50 Sfrs, 30 FFRS)

Na osnovu mišljenja Republičkog sekretarijata za kulturu broj 413-77/72-03 i Službenog glasnika broj 25/72 ovo izdanje oslobođeno je poreza na promet

Sadržaj

6/ razglednica iz Londona
velikani na rasprodaji

8/ naš test
„kanon" protiv „epsona"

10/ računari u izlogu
o „komodoru" sve najlepše

13/ računari u razgovoru
programer koji je hteo da promeni svet

17/ periferijska oprema
svetlosna pera

20/ umetnost programiranja
haker na usijanom limenom krovu

22/ smešna strana računara
računaranje i računarsko razgovaranje

23/ psiho test
pirati i obožavaoci

25/ računari u akciji
baza podataka

29/ programi koje treba imati
na daj se, Ines!

30/ računari u poslovnoj primeni
mali računari u velikoj privredi

32/ biblioteka programa

56/ test sa zadrškom
prama svecu i drajv

58/ hajde da se igramo
šest igara za „komodor"

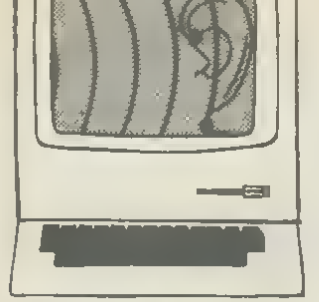
60/ majstorija na računaru/„komodor"
putovanje u provincije

64/ programiranje u bejziku
funkcije, potprogrami, procedure

68/ grafika na računaru
kako to radi „amstrad"

71/ majstorija na računaru/„spektrum"
kodovi iz rukava

72/ put u središte rom-a (4)
lične stvari bejzika



šta ima novo

„Galaksija plus"

Godinu i po dana nakon pojave računara galaksija u prostorijama Zavoda za učila i nastavna sredstva u Beogradu održana je demonstracija galaksija plus koja je takođe nastala kao proizvod saradnje Zavoda Elektronika inženjering iz Zemuna i časopisa Galaksija. Autori hardverski i softverski proširene galaksija plus su Milan Tadić i Nenad Dutnić koji su čitaocima poznati već po tome što su galaksiju obogatili grafikom visoke rezolucija.

Već i na prvi pogled nova galaksija razlikuje se od stara kutija u koju je računar smešten našlo je veća i na njoj sa nalaze izvodi za priključak ispravljača kasetofona televizora, monitora i audio pojačivača a ugrađen je i konaktor za priključenja paralarnih jedinica odnosno jedinica za priključenja novih jezičkih procesora ili aplikativnog softvera u ROM-u. Profesionalna tastatura inača proizvod IEVT iz Ljubljana sadrži i posebna tastera za č ć ž i š što takođe predstavlja novinu.

Galaksija plus ima ugrađena dva ROM-a U prvom ROM-u kapaciteta 8 K smešlan je bejzik interpretar i standardni Z80 asambler dok se u drugom (kapaciteta 2 K) nalazi ekranski editor. Osim toga galaksija plus sadrži 48 K dinamičkog RAM-a. Pored proširenja memorija i generatora zvuka tu je i fina grafika (256 – 208 tačake) sa mogućnošću prikazivanja više slika koja se mogu smenjivati u svakom frame intarvalu ili jedna virtuelna slika koja se prikazuja sa mekim pomeranjem kao i postajanja gašenja i testiranja svaka tačka.

Treba spomenuti i softverski izmenjiv set karaktera (mala slova matematički simboli grčki alfabet itd.) kao i mogućnosi prikazivanja više setova simultano na ekranu. Od priključaka novine predstavljaju pobojšani kasetni ulaz/izlaz audio izlaz i dva paralelna osmobitna input/output porta.

Nama sumnja da nova verzija galaksije predstavlja znatno poboljšanja u odnosu na prethodnu. Za sada postoji samo nekoliko primeraka ovog računara ali bi trebalo očekivati da će i galaksija plus uskoro naći svoju primanu pra svega u osnovnim i srednjim školama. U sledećem broju objavljujemo detaljan prikaz nove galaksije.

Istočne jabuke

Prema Mikroszamitogep magazin u (Mikroračunarski magazin) časopisu za madarske ljubitelje računara u Madarskoj se 1984 tak stidljivo razmišljalo o razvoju računara koji bi bio kompatibilan sa eplom II.

Medutim, u Bugarskoj su se za taj pravac odlučili mnogo ranije pa su već 1982 ponudena tržišlu dva mikroračunara — IMKO-2 i Pravec-82. Ova dva računara su bila namenjena pre svega školama a proizvedeno je više hiljada komada. Prošle godine svelo dana su ugledala dva nova i snažnija modela — Pravec-8B i Pravec-8M. Ova dva računara uz svoj imidž imaju i elikatu za profesionalnu namenu.

U SSSR proizvodi sa mikroračunar AGAT takoda kompatibilan sa eplom II ali baziran na sovjetskoj tehnologiji tj. na komponentama koja su proizvedene u Sovjalskom Savezu.

U Madarskoj pak pod imanom AX-II u 1984 pojavljuje se mikroračunar koji je napravilo Omladinsko istraživačko društvo.

Ipak glavni događaj da ne kažemo događaj sezone bice pojava prvih 100 koma da mikroračunara (takode kompatibilnih sa eplom IIa) po imanu IVEL Z-3.

Ovaj mikroračunar u srcu ima jedan Z80 i dva Rockwell 6502 mikroprocesora. Opremljan je epi DOS-om 3.2/3.3 i uz sve to i CP/M i USCD. U sklopu mikroračunara je disk jedinica sa dva drajva.

Ipak glavno osveženja ostavljeno je za kraj. Ovaj mikroračunar se proizvodi u kooparaciji pri čemu Videoton Rt radi terminalni deo a računarski deo jugoslovanska firma IVASIM !!!

Računaru se prognozira veoma svetla budućnost pri čemu mu se najavljuje cena od oko pola miliona forinti što prama madarskim kursnim listama iznosi 1.9 miliona dinara.

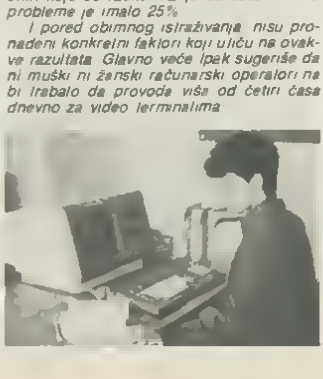
I. Nador

Računari i žene

Stara i veoma rasirana rasprava o tome da li računari i računarski sistemi utiču negativno na čovekovo zdravlja obogaćena je ovih dana sa dva nova priloga — iz Japana i Velike Britanije.

Prema saopštenju Glavnog veca japanskih Trajd Uniona više od jedne trecine žena koje rade za računarskim terminalima je imalo problema za vreme porodaja. Ispitivanjem je bilo obuhvaceno 250 žena koje su zatrudnile ili se porodile za vrama ili nakon rada na terminalima. Od trudnih žena koje su radile za terminalom 6 časova ili duže dva trecina je imalo problema sa porodajem. Od onih koja su radila tri do četiri časa, problema je imalo 46% a od onih koje su radila manja od sata dnavno probleme je imalo 25%.

I pored obimnog istraživanja nisu pronadeni konkretni faktori koji utiču na ovakve razultata. Glavno veće Ipak sugeriše da ni muški ni ženski računarski operatori na bi trebalo da provode više od četiri časa dnevno za video terminalima.



Gotovo istovremeno Britanska vlada je u specijelnoj izjavi, koja se tiče tog istog problema saopštila da ne postoje konkretni dokazi da video terminali negativno utiču na tok trudnoće i porodaj. U specijalizovanom izveštaju vladina agencije koja se bavi tim problemom takode stoji da su nagomilavanja broja podataja među ženama koje rade na terminalima verovatno rezultat statističkih anomalija.

Mada u ovoma izgleda ima mnogo više dima nego vatre. Bilo bi interesantno videti da li se i kod nas neko brine za problem trudnih žena koje rade za terminalom. Istina kod nas nema toliko računara da bi to bila rašireno pojava ali zdravlje je zdravlje tim pra što se zna da kod nas na unošenju podataka prako terminala rada gotovo isključivo žene.

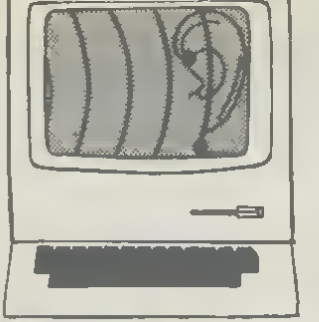
B. Đakovic

„Komodorovo novo ruho

Oni koji su bar jednom videli velike računarske sisteme kako rade sa hard diskovima znaju kakva je razlika izmedu njih i običnih disk drajvova. Sada vlasnici „Komodora imaju to zadovoljstvo da kažu da i njihov računar ima neka odlike velikog računara. Mala računarska firma COMPUTER SPECIALTIES napravila je prv hard disk za komodor 64 sa oznakom CSI ST 10C. Disk je kapaciteta 10 megadajta što predstavlja oko 50 000 strana gusto kucanog teksta a brzina prenosa podataka je prosto fantastična — za nekoliko stotih delova sekunda popunjava se čitava memorija od 64K. Disk koristi serijski ili IEEE bas za prenos podataka. Jedina ali za nas sasvim dovoljna mana je — cena. Disk košta lačno — 1 595 dolara ili oko 479 000 dinara. Ako vas ovo nija ohladilo za detaljnije informacije možete pisati na adresu CSI P.O. 1718 Melbourne Florida 32902.

Interfejs za povezivanje „komodora 64 sa eplom II+ omogućava vlasnicima komodora da koriste sav posložaci softver i hardver za epi II+ a toga ima dosta. Mogu se koristiti i eplovi diskovi i sve ona silne kartice (slotovi) koji su od „epla napravili najrašireniji računar u svetu. Ni ovaj interfejs nije bez ozbiljnih mana — košta oko 600 dolara što je pri trenutnom kursu dinara zaista prilično mnogo. Za sva informacija obratila se na MIMIC SYSTEMS INC. 1112 Fort ST. Victoria B.C. CANADA V 8 V 4 V 2.

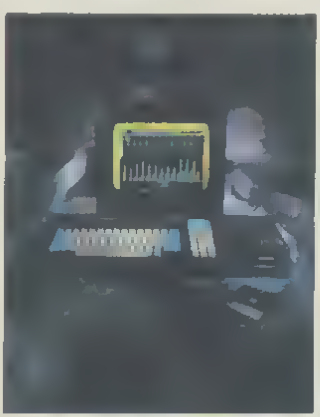
Treća, za nas možda i najinteresantnije novost je program koji na „komodoru 64 simulira ZX spektrum. Pokazuje sa da je komodor 64 dovoljno brz računar da može da simulira rad nekog manjeg računara što može biti vrlo interesantno. Ljudi koji su napravili ovaj program tvrde da će svi bejzik programi raditi sasvim normalno čak i istom brzinom. Uz to pošto i „komodor i spektrum imaju RS-232 možete koristiti i neke spektrumove dodatke kao na primer ZX Interfejs II. Jedino traba probati kako ovaj program radi sa nekim ozbiljnijim bejzik programima koji se oslanjaju na grafiku. Ako sta zainteresovani za ovaj interesantan program obratite se bilo kom beogradskom piratu. Program vec kruži Beogradom. Živeli MALI pirati!



šta ima novo

Novi „amstrad"

Pojava [illegible] modela tvrtke Amstrad — sarenog [illegible]dela CPC 464 — i njegove ozbiljnije obostrane zapadno-njemačke kopije tvrtke Schneider počela je dugo očekivana prekretnica u razvoju integriranih sistema kucnih racunala. Za svega 1400 DM moguce je nabaviti računalo s ugradenim kazetofonom ta trio kvalitetnim RGB monitorom a za 900 DM istu konfiguraciju s zelenim monitorom. Na tržištu je ovo računalo manja od godina dana a vec ima široki krug konsnika. Tvrtka Amstrad svjesna svog poslovnog uspjeha na tržište izbecuje novi modal CPC 664 koji za razliku od prathodnog modela ima ugradenu diskjedinicu umjesto kazatofona. Istovremeno konstruktori amstrada najavljuju nove modela koji ce biti sottverski kompatibilni s prathodnim modalima a zna se što to znači za plasman i konštanje proizvoda ove tvrtka te buduca korisnike — obilja višestruko iskoristivog sottvera na izvanrednom hardveru.



Cijena novog računala ostaja u stilu „svega 400 funti" što je i aftinija od ranijeg CPC 464 i disk jedinice. Ugradani CP/M operativni sistem osigurava veliku sottverska podršku, pa je i to prilog rantabilnosti ovog računala.

Računalo CPC 664 radi s procesorom Z 80A (4MHz). Memorija je i dalje 64K od kojih je 41K upotrebljiv za korisnika. Računalo koristi i izvjesna mogucnosti preklapanja. ROM memorije koja ima veličinu 32K, u kojoj je sadržan pobolšani Locomotiv bejzik i operativni sistem. Takva konfiguracija memorija omogucuje vrlo složeno programiranja.

Tastatura je ukusnija obojena sivo — svijetlo plave (neke tipka). Sadrži 74 tipka standardnog QWERTY rasporeda u svamu ista kao i kod ranijag Amstradovog modela 32 tipke mogu biti posebno definirane od korisnika.

Zvuk sa računala se generira zvučnim čipom AY-3-8192. Na raspolaganju stoje tri kanala s rasponom od 8 oktava. Svaki kanal može biti nazavisno namješten na odredeni ton i amplitudu odnosno na ljavi desni i centralni signal što sa može koristiti kao stereo ton prako posabnog izlaza. Jacina zvuka iz ugradenog zvučnika je promjenjiva a predstavlja miksani monoauralni izlaz.

Uz računalo je moguca koristiti mnogo dodatnog hardvera. Oni naviknuti na kazetofon mogu ga priključiti na poseban port. Amstradov matrični printer DMP-1 novi džojstik JY-2 ta druga disk jedinica FD 1 predstavljaju osnovnu dodatnu konfiguraciju. Naravno moguca je koristiti i proizvoda drugih tvrtki. Oni koji na žala nabavljati monitor i dalja imaju na raspolaganju modulator MP-2 prako kojeg mogu računalo priključiti na obican kolor TV-prijamnik. Inača računalo ima izvedani RGB i kompozitni video izlaz.

Locomotiv bajzik je pobolšan ali kompatibilan s bejzikom na CPC 464. Glavna prednost pobolšanog bajzika su vrlo brze area fill naredba odnosno pobolšanja sa uglavnom odnose na komanda za rad s diskom. Takodar postoji i nova interapt sabrutina vezana uz instrukcije AFTER i EVERY.

Grafički modovi su ostali isti kao i na ranijem modelu.

Priključnica ima ukupno 8 za printer (paralalni Centronics), za dodatnu disk jedinicu joystick RGB i kompozitni video signal kazatofon stereo zvučni izlaz ta priključnice za napajanja (5V i 12V). Napajanja sa dobija sa monitora jer je u njamu ugradan ispravljač.

Vec latimičnim pregledom tahničkih karakteristika se vidi da je jedina bitna razlika izmedu CPC-a 664 i CPC-a 464 samo ugradena disk jedinica.

Ona odgovara Hitachi/Panasonic standardu od 3 inča (1 inč=2,54 cm). Cijeli sistam računala je koncipiran da može podržavati maksimalno dvija disk jedinica.

Operativni sistam na Amstradu je AMSDOS ali i CP/M. Moguca je koristiti i jazik LOGO. AMSDOS je operativni sistam koji proširuje Locomotive bajzik a omogucuje spramanja podataka slično kao kod spramanja na traku. Upotraba CP/M-a omogucuja korištenja izuzetno valika programska podrška. Organizacija diska je takva da AMSDOS i CP/M podržavaju tri različita formata diskova. SYSTEM DATA only ta IBM format. Selakcija formata je automatska na zahtjav diska. Sva tri formata koriste isti „framework" ali različitu konfiguraciju saktora. Zajedničko za sve formate je 512 bajtna valičina saktora, 40 STAZA te umetanje saktora u omjaru 2:1.

Konstruktori amstrada su sa opredalili za standardni SYSTEM format. 2K sa koristi za vodenja a 9K je razavrivano za sistem. Ima 9 sektora po stazi i 2 rezavrirana staze za CP/M. Ima mogucnost pohrana 169K podataka.

DATA only format koristi sve staza za podatke — 2K je rezavrirano za vodenja a ima 9 sektora po stazi i kapacitet 178K.

IBM format je sličan kao format koji koristi CP/M na IBM PC-u. 2K se koristi za vodenje a 4K je rezavrirano. Postoji 8 sektora po stazi a 1 staza je rezervirana. Kapacitet je 154K.

Mikro diskete koja sa koriste imaju kapacitet 360K (2×180K) a puno su skuplja od npr. 5,25 inčnih.

Dakle osnovna razlika starog i novog amstrada je ugradena diskjedinica a može se priključiti i dodatna pa čak i ona od 5,25 inča. Amstrad CPC 664 predstavlja do sada najsretnije koncipirano računalo a ako se u obzir uzma njegova cijena bolje nesenja za korsnika računala u našim uvjatima nema.

Z. Vistricka

Deset megabajta na jednoj pločici

Na nedavno održanom sajmu za IBM PC računare u Njujorku jedan od najzapaženijih proizvoda bio je tvrdi disk (hard disk) sa kapacitetom od 10 megabajta koji je napravljen tako da izgleda kao periferna kartica koja sa diraktno priključuja na jedan od slotova na IBM PC i njamu kompatibilnim računarima. Dimanzije su mu: dužina 33 cm, šinna 10 a debljina svega 2,5 cm.

Ovaj proizvod koji nosi ime HARD-CARD predstavlja rezultat 20-tomesečnog rada kalifornijska kompanija Plus Davelopment i japanske firma Matsushita. HARD-CARD ce biti puštan u prodaju u Amarici u oktobru ove godina po cani od 1 095 dolara i proriče mu sa sjajna budućnost. Višemesečno testiranja ovog tvrdog diska na računarima Compaq (PC kompatibilnim) izvršeno u poznatoj fabrici džinsa Levi Strauss u San Francisku dalo je odlična razultata. Ibeju ga ovaj proizvod dao je pionir kucnih računara i video igara legandarni Nolan Bushnel (Bushnall) osnivač firma Atari. Kada smo već kod Nolana da spomenemo da



je on posla optimističkog starta digao ruke od proizvodnja parsonalnih robota zbog slaba prodaje Alf Nolan na posuslaje — upravo je sklopio ugovor sa američkim automobilskim gigantom General Motors za isporuku specijalnih računara za kontrolu motora i orijentaciju vozila

A Zgorelec

Ejkorn nije umoran

I pored svih nevolja kroz koje je Acorn protazio i protazi njegova zvanična softverska firma Acornsoft po svemu sudeći dobro posluje Očekuje se da se na ovogodišnjem letnjem sajmu časopisa Acorn User Acornsoft predstavi nekoliko novih i veoma interesantnih naslova

Radi se pra svega o novoj verziji njihovog tekst procesora View koja nosi oznaku 3 0 U prošlim računarima smo predstavili verziju 1 4 ovog programa i napomanuli da verzija 2 1 još nismo imali prilika da vidimo U međuvremanu je View 2 1 pristigao u Jugoslaviju ali sa, kao što to obično biva odmah zatim pojavila nova verzija

Acornsoft će svakako reći da View 3 0 rezultat firmina želje da stalno unapređuje svoje produkta ali izgleda da nije baš tako View 2 1 je pokazao nekoliko veoma ozbiljnih bagova koji se odnosa na konfiguracija penferija Da bi bio kompatibilan sa američkim verzijama BBC B koji imaju nekoliko redova teksta na akranu manja View 2 1 po inicijalizaciji proverava sistemsku promenljivu HIMEM u kojoj je upisana adresa prvog bajte video memorije, ako ona nije na nekoj od vrednosti predviđenih za evropski BBC računar radi sa manje redova na akranu Na nesraću naki vlasnici BBC B su se opremili RAM tablom uz pomoc koje im sva 32 Kb osnovne memorije ostaju slobodna bez obzira na grafički mod Kod njih je HIMEM na &8000 (kraj RAM-a) pa se View pogrešno inicijalizuje!

Osim ispravka ovoga baga autori View 3 1 su napravili jedan odlićan potez vlasnici 6502 dodatnog procesora su do sada morali da kupuju takozvani Hi View jer sa običan View u memoriju dodatnog procesora prepisivao na adresa &8000-&BFFF što znači da je za tekst ostajalo memorija kao u slučaju kad dodatnog procesora nema dok je prostor od &8000-&FFFF zvijao prazan View 3 0 je dovoljno intaligentan da se sam relocira prepravljajući svoje apsolutna skokove i apsolutna adresiranja Na taj način se takst koji pišete može prostirati na preko 40 K u bilo kom grafičkom modu

Samo za vlasnike 6502 dodatnog procesora Acornsoft je pripramio i Elitu 2 Nova verzija je slična staroj ali je u koloru i daleko brža a dodata su i naka nova oružja i vrste oprama kojima treba opremiti Cobru Ako imate BBC B ova igra je odlićan razlog da dokupite drugi procesor!

Drugi procesor kada ga već kupite može da posluži i za ozbiljnije stvari Acornsoft je pripramio diskatu sa Fortranom 77 i Paskalom koji su potpuno implamentirani po vazecim standardima (računajući komplaksne promenljive i funkcije u fortranu) Cana diskete je prava sitnica — svega 300 funti! View 3 0 košta samo 70 funti a Elita naravno 17 95

Dejan Ristanovic

Crne vesti za pirate

Piratima je ovih dana i lakše a i sve teže da rade

Činjanica je da pirati sve lakše kopiraju tude programe zahvaljujući pored ostalog i sve savršenijim tehnickim sredstvima koja im pomažu Jedno od takvih sredstava je i Ivešam Mirkos Interfejs III (Evesham Micros Interface III) Upotreba je sasvim jednostavna IMI III sa priključi na računar u računar se unese neka igra ili nešto slično pritisna se dugma na vrhu IMI III i čitav sadržaj memorija računara odlazi na mikrodrajv izbegavajuci bilo kakvu zaštitu koja se može staviti na traku ili disketu To je dakle lepa vest za naše pirata

Drugu činjanicu da je piratima sve teže da rade možda bi trebalo prelormulisati u piratima će uskoro (u Velikoj Britanji) biti jako jako teško da rade Predvida se da ce ove godina biti usvojen takozvani Vilijem Pauel kompjuter kopirajt predlog zakona, kojim se zabranjuja izrada bilo kakvih kopija izdatih programa osim za istraživanja i proučavanja Tako ca se i u softverskim izdanjima prihvatili pojam kopirajta Kazne za kršenja ovog zakona ce biti jako oštra što bi trebalo da sasvim određeno utiće na broj pirata u Engleskoj Znači crna vesti za pirate u Britaniji ali bi mogla biti loše i za domace pirate ako sa naši zakonodavci budu ugledali na svoja britanska kolege

B. Đakovic



IBM napusta Epsona

Izgleda da su američki proizvodaci kucnih računara sa zakasnjenjem shvatili da se na stampacima moze razraditi dosta novca i da je to jedna od najpopularnijih perferijskih jedinica Dok su oni pre 5—6 godina dominirali tržištem (Centronics itd) u poslednjih nekoliko godina su dozvolili da ih Japanci preteknu i da zauzmu čak 80% amaričkog tržišta koje sto se liče stampača godišnja vredi 7 miljardi dolara

Sada medutim i IBM i Harox najavljuju ulazak na to tržišle sa vlastitim štampačima!

IBM planira jeftiniji matrični stampač nazvan proprint dizajniran za tri različite brzine od 40 100 i 200 slova u sekundi Cena u britaniji ce biti 499 funti Xerox uskoro pušta u prodaju laserski štampec velike brzine jer se takvi štampači zbog kvaliteta otiska i brzog stampanja sve više traže

IBM takode napušta prodaju Epsonovih stampača za PC računar pod svojim imenom zbog slaba prodaje Oni su naime bili nešto skuplji od indentičnih Epsonovih štampača a samo tri magična slova na kutiji nisu bila dovoljna Ovaj korak IBM-a neće mnogo pogoditi Epsona jer sa njegovi računan odlično prodaju i sada zauzimaju čak 30% amaričkog tržišta

A Zgorelec

Koga ne boli glava

Izgleda da ipak nije sve tako crno u svetu mikroračunara pogotovu za proizvođače i izdavače softvera za IBM PC i njemu kompatibilne računare Na nedavno odrzanom PC sajmu u Njujorku mogto je da se vidi nekoliko veoma interesantnih proizvoda pogotovo sa područja softvera

I londonski PC sajam bio je veoma uspešan — preko 200 izlagaća i oko 15 hiljada posetioca Jedan od trendova uočljiv se tih sajmova je da gotovo svaki veći proizvođač nastoji da proizvede mali prenosni računar sa ravnim ekranom od tečnog kristala (liquid crystal)

Za sada se takvi računari medutim još uvek ne traže

Na području softvera mnogi izdavači su se trudili da pokažu da su njihovi integrisani programi korisniji od poznatih jednonamenskih Ali bar za sada kupci izgleda više veruju jednonamenskim programima, koji su jednostavniji i rade odlično nego komplikovanim integrisanim koji samo zadovoljavaju

Smatra sa da ca odlična budućnosi imati programi specijalno pisani za pojedina struka advokata inženjera farmera itd

Na ovim izložbama bilo je i više izlagaća koji smatraju da je sledeći trend da upotreba računara buda što jednostavnija pa se mnogi utrkuju da naprave računar što sličniji makintošu sa mišom i prozorima

Izgleda medutim da je glavni razlog slabe proda kućnih računara to što potencijalni kupci na vide kako im oni mogu koristiti i učiniti život lakšim

A Zgorelec

Razglednica iz Londona

velikani na rasprodaji

Sreće u industriji kućnih računara verijivije je, rekio bi se, ček nego i u samim igrama na sreću. Vrtoglavi usponi i još strmogliviji padovi ovde se doživeju bukvalno preko noći. Perspektivnim preduzećima koje su nepročec otišle pod led ni broje se ne zna, a žrvnje u „točku sreće" nisu poštedene ček ni najveće imena. Za nepunih pole godine na rasprodaji su se našle i dve najveće žive legende iz domene kućnog računarstve. Pre nekoliko meseci se evog trone u „Apple"-u svrgnut je Stiven Džobs, a sede je slične sudbine zadesile i legendarnog Klajve Sinklera, oca evropske mikroelektronske revolucije. Prodajom svoje kompanije, on je i poslednje veliko uporište biznisa u rsčunerskoj industriji predeo u ruke profesionalaca.

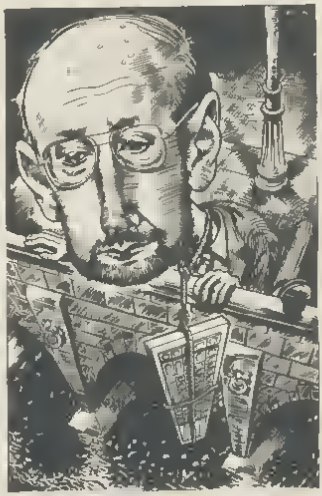
Keo što se i očekivalo nije trebalo čekati dugo de se spesi najsvetlije ima britenskog mikroračunerstve U akciju spasavenja ukjučile su se i poznate banke i finenciske institucije i uticajni politićari Spasilec medutim nije došao iz elektronske ili kompjuterske industrije već iz sveta biznisa

Vitez na belom konju

Sinklerov beli vitez kako ge nazivaju ovdešnji listovi, je Robert Maksvel (Makawell) 60-godišnji veome uapešni eli često i kontroverzni biznismen Mekavel, koji je po poreklu Čeh, došao je posle rate u Britaniju i uspeo je da za ciglih četrdeset godine postene jedan od najisteknutijih poslovnih ljudi ne Ostrvu On je vlasnik jedne od najvećih britenskih izdevečkih kuća za naučno-tehničku literaturu — Pergamon Press Pre nekoliko godine uapeo je de spase od bankrotstva najveće britensko štemparsko preduzeće — BPPC — i pretvori ge u profitabilnu kompeniju Prošte godine postao je i vlasnik jednog od najtiražnijih ovdašnjih dnevnika _Dejli Miror (Deily Miror) jedinog dnevnike koji podržava leburističku stranku (Mekavet je i sam bio više godine leburiatički poslenik perlemente) On je takode vlasnik i novog engleskog prvoligeša time Oxford city

Keda je u maju Meksvel sazneo da je Sinkler u finensijakim poteškoćame i kede je vidio de nema sigurnog kupce, odlučio je da stupi u akciju jer keko on kaže, „bile je u pitenju čast necije Za ulog od 12 mitione funti i uz preuzimanje nekih kreditnih obaveza, on je kupio 80% akcija i kontrolu nad Sinkler Riserćom Finenaijake institucije zadržavaju 10%, e semom Sinkleru tekode ostaje 10% njegove bivše kompenije uz ulogu savetnike i počasne titule doživotnog predsednika

Ove dobre vest ze Sinklere izazvela je oduševljenje medu vlesnicime Sinklerovih računere u Britaniji keo i u drugim zemljama Robert Meksvel se slevi kao spasitelj kućnog računerstva No, Meksvel na sedi na lovorikeme — on smetre de Sinkler mora de poveća izvoz po sveku cenu! U



Greškom u svetu velikog biznise Klajv Sinkler

svom stilu, de je vreme novac, on je za nekoliko dene obišeo nekoliko zemalje latočne Evrope u trgovečkoj misiji Poznet od ranije po odličnim poslovnim vezame sa socijelističkim zemljama, njege su primile i najviše ličnosti u nekim zemljeme Tako je preme plaenju londonskih listove on rezgovorao I se poljskim premijerom Jeruzelskim i, keko se ovde tvrdi Poljska bi uskoro trebalo de uveze 60.000 „spektruma+ koji će se u početku prodeveti u radnjame Pewex gde ae zepadni proizvodi mogu kupiti semo za devize Nevodno, i Bugarska i Sovjetski Savez pokazuju interes za Sinklerove računere i pregovori su u toku Sinkler je nedevno pokazao i nejnoviju verziju „spektrume+ se ćirilićnom testeturom Izgleda de su Sovjeti doste zainteresovani i ze QI pogotovo pošto je početkom jule britanake vlede skinule zebranu sa izvoze sevršenijih računera u zemlje latočne Evrope

Quo vadis, Sinkler?

Izgledi de ovi finansijski probtami i nisu mnogo uzbudili ser Klajva On se i delje pojavljuje na sajmovima seminerima i televiziji I delje priče o svojim idejama o revolucionarnijoj petoj generaciji računare, na kojima sade radi Samo je pitanje da li će ga finensijake institucije i banke i dalje podržaveti novčano Jer Sinkler je nekime vec postao dosaden — ponaša se kao da igra ulogu restresenog naučnika koji je greškom ušeo u svet velikog biznisa i velikog novca To njegovo tapkanje u avetu biznise stajalo te banke u poslednjih desetak godina i dosta novaca Bilo je tu poslovnih grešaka sa kalkulatorime i digitelnim časovnicime melim prenosnim televizorima električnim vozilom — da nebrojimo aemo nekoliko od njih meda su sve to u svoje vreme biti revolucionarni proizvodi koji bi možde uz bolji marketing i bolje poslovno vodenje imeli drugu budućnost

Računarski Kolčicki

I pored tmurnih vesti izgleda ipak de industrija mikroračunare polieko izđezi iz krize Najgore kao da je prošlo Poznati komentetor sa toga područje Rej Hemond (Rey Hemmond) deje sledeću prognozu za šest meseci

***Sinkler** — ove godine neće biti novog kućnog računere Početkom sledeće biće lensirena prenosna verzija sa ugrađenim ravnim ekranom i mikrodrajvime Spektrumu+ se neće sniziti cena već će neki prodavci davati besplatni softver Spektrum + će i u nekoliko sledećih godina biti najvažniji kućni računer*

***Ejkorn** — ništa novo osim meio sniženje cene za BBC računar*

***Ametrad** — još jedan novi model ove godine 16-bitni računer programski kompatibilan sa prethodnim modelime*

***Komodor** — neda se pokažu u modeli C128 koji je u velikoj meri kompatibilan se popularnim modelom C64*

***Atari** — previše se obećeve i biće dobro eko se ispuni i pola Veliko sniženje cena sadašnjim modelima*

Sinkler je nedavno dao i intervju nedelj niku MicroScope gde tvrdi da on nikada nije ni želeo da bude biznismen (a zašto to nije rekao ranije?) već da je najsrećniji kada radi u laboratoriji. Sinkler tvrdi da je srećan što je došlo do sporazuma sa Maksvelom i da se nada da će Sinkler Riserč stati na zdrave noge, pošto će ga voditi pravi poslovni ljudi. On će biti plaćen za svoj posao kao savetnik u kompaniji a zadržao je i pravo da samostalno radi na petoj generaciji računara, mada Maksvel želi da se to radi u okviru Sinkler Riserča i podružnice Matlab u Kembridžu. Ukoliko Sinkler ne želi da se pridruži radu na inteligentnim računarima, Maksvel namerava da okupi ekipu svojih stručnjaka. Sinkler bar za sada ne želi da otkrije svoja planova — samo kaže da se nada da neće doći do spora sa Bobom Maksvelom.

U ovoj zbrci oko spasavanja Sinklera, malo je nezapaženo prošla vest, koja je došla iz Sinklarova podružnica za naučna istraživanja Matlab da je uspešno konstruisan visokointegrisani čip koji može da prauzma utogu svih čipova u našem računaru. Probna proizvodnja bi trebalo da počna već sledeće godine.

Olivetti diže ruke?

Drugi najistaknutiji predstavnik britanske mikroračunarske industrije firma Ejkorn (Acorn) i dalje ima puno problema. Kao što smo javili pre nekoliko meseci italijanski gigant Olivetti je kupio polu deonica Ejkorna i tima, u stvari, preuzeo aktivnu kontrolu. Ali i pored investicija od preko 10 miliona funti koliko je plaćano za ta deonice, koja se sada koristi za otplatu dugova, finansijski problemi se i dalje gomilaju. Zastarali i preskupi model BBC se slabo prodaje a najnovija varzija BBC+ nije doživela komercijalni uspeh tako da veoma malo novaca ulazi u kasu Ejkorna. To je primoralo ovu firmu da potraži kupca za svoja podružnica, među kojima je i najvažnija softverska kompanija Ejkornsoft.

Najveći interesant za Ejkornsoft je British Telecom kompanija koja je vlasnik telefonska mreža u Britaniji i koja ima vlastitu uspešnu softversku firmu Firebird. Za oko 1 milion funti British Telecom bi trebalo da postane vlasnik preko 1000 odličnih programa, kao i gotove roba čija je prodajna vrednost blizu 2 miliona funti. Nota svota je tek kap u moru Ejkornovih finansijskih poteškoća. Izgleda da Olivetti na žali da neograničeno otvori kesu i zato je došlo do ponovna suspenzije Ejkornovih deonica na londonskoj berzi a banka fraže nove mogućnosti za otplatu dugova. Prama vestima iz londonskog Sitija, postoji čak i mogućnost da Olivetti potpuno otpiše svoju investiciju od 10 miliona funti i digne ruka od Ejkorna. Drugo rešenje za Olivettija je da i dalje sipa novac u vreću bez dna ili da prode Ejkorn nekom trećam investitoru. Ali da li postoji još neki Bob Maksvel koji je zainteresovan za pune magacine robe koje sa sve slabije traži? Sudbina Ejkorna sada najviše zavisi od sposobnosti bankara da osiguraju dodatne kredite.

Andelko Zgorelec

7/velikani na rasprodaji

Naš test „kanon' protiv „epsona'

KAGA TAXAN KP810/CANON PW1080

Prikazujući poznate Epsonove modele FX80 i FX100 u „Računarima 2", zaključili smo da se radi o najboljim štampačima u svojoj klasi, koji predstavljaju dominirajući svetski standard. U međuvremenu, ostale japanske firme nisu sedele skrštenih ruku — na tržištu se pojavilo nekoliko štampača koji su, i pored znatno niže cene, nudili potpunu kompatibilnost sa Epsonom FX80. Štampači „Kaga" (Kaga) KP810 i „Kanon" (Canon) PW1080A, uklapajući se u ovaj trend, nude i nešto novo. NLQ (near letter quality) mod koji se po kvalitetu otiska opasno približavaju štampačima sa lepezom. Obzirom da se „Kaga" i „Kanon" razlikuju jedino po ceni (drugi je dvadesetak funti skuplji), odlučili smo da im posvetimo zajednički prostor u „Računarima".

Kada prebrodite sva carinska problema i donesete paket težak osam kilograma do kuća očekuja vas uzbudljiv zadatak računar je obično lako otpakovati i priključiti, disk nešto teže a štampač predstavlja pravi problam osiguran je gomilom šrafova gumica, lepljivih traka i drugih sprava koje tesfiraju vaše strpljenja. Kaga je dobra potvrda ovog pravila kada sa uvarite da vam se u paketu nalazi štampač kabl ručica za okretanja papira traka i naravno uputstvo za upotrebu moraćete najpre da odšrafljujete dno printara od dna kutija zatim da polomite nokte odlepljujući salotejpe oslobodite glavu snažne gumice koja joj na dozvoljava da se pomeri montirate utikač na kabl (ako vam štampač stigne u subotu popodne a u kući nemate ni jedan šuko priključak, sačekačete do ponedeljka!) i montirafe traku i papir.

Poslednja dva operacija su najodgovornije traka sa nalezi u kaseti koja je drukčija od standarda koji je nametnuo „epson" (akuplja je dvadesetak panija) i na treba je provući između limenog zaštitnika i papira već između gtave i limenog zaštitnika ako poput autora ovoga tekata uradite prvo sva će lepo raditi ali će traka izlaziti iz ležišta kada god naida na perforaciju. Pra umetanja papira pažljivo proverite da li su oba praklopnika na mastima koja odgovaraju papiru koji koristite, kaga i „kanon' mogu da pišu kako ne standardnoj perforirenoj hartiji tako i na rolnema papira ili pojedinačnim listovima. Posebno je teško umetati obične listove glavni preklopnik će, najpre moreti da bude u položaju Paper Set zetim u položaju Pin Feed e onda će zeuzeti svoj konačni položaj Friction. Ze eksperimentisanje sa printarom čete jasno odebrati perforirani papir ili eko možete da je nebavite rolnu.

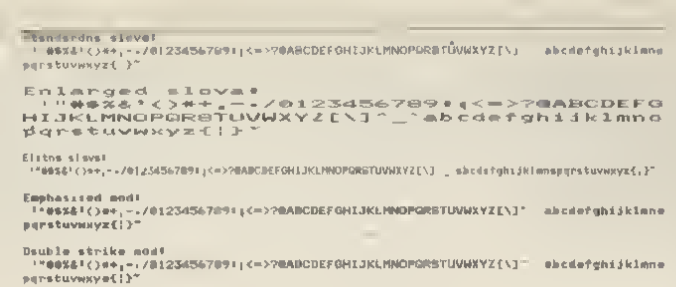
Pošto ste sve pripremili printer će testirati samoge sebe pritisnite senzor LF (da dobro ste pročitali umasto epsonove tri prekidača „kaga" i kanon imaju isto toliko senzora koji nakim slučajam, imaju potpuno iate funkcija) i, držeci ga pritisnut uključite štampač. Ukoliko je sva u redu trebao bi de se začuje kratak bip e zatim započne ispisivanje teksta za nas je to bila prilika da prikažemo standardni sat karaktera baz potreba da pišemo bajzik programi.

Oiskretna veza

Kaga i kanon su, u osnovnoj verziji opremljani Centronics interfejsom što znači da ih možete povezati sa amstradom BBC-jem eplom IBM-om i mnogim drugim računarima. Što se komodora i spektruma tiče možeta da birate između kupovina RS 232 interfejsa koji se ugrađuje u štampač ili Centronics interfejsa koji se povezuje sa računarom (ukoliko imate spektrum pa se odlučite za prvo rešenje moraćete da kupite i interfejs i). Da spasamo redakcijski talaton nismo nigde videli oglas u kome bi se bilo koji od ovih štampača nudio sa interfejsom RS 232 a baz centronicsa. Razlog za to je varovatno činjenica da Centronics opcija i ne zahteva nake posebne komponente to je prirodni način na koji štampač prima podatke. Ako želite serijsku komunikaciju doplatite 50 funti za interfejs i dobićete sa njim printar bater od 3 kilobajta.

Kada najzad povežete računar sa štampačem (pazite kabt nije uračunat u cenu!) možete da testirate sve tipove slova koje kaga (kanon') nuda za to vam je potreban program poput onoga sa slike 2 ispod koga su prikazani razultati njegovog izvršavanja. Pažljiv posmatrač će primetiti de su korišćeni modovi (Standard Enlarged Condensed Condensed enlarged Elite i Elite enlarged) po imenu i po kontrolnim kodovima koji ih aktiviraju identični sa „epsonovim". Vrlo pažljiv posmatreč će međutim primetiti da slove nisu potpuno identična epsonovim što se najbolje vidi kod standardnog seta karakteri su za nijensu niži i oblici nekih su promenjeni (pogledajte na primar nulu!). Da li razloge za promene treba tražiti u zakonu o zaštiti autorskih prava ili u hiru kanonovih dizajnera teško je reći, u svakom slučaju stvar je ukusa koja će vam se slova vise dopasti pri čemu je nesumljivo da je njihov kvalitet jednak printer ih crta na matrici 11'9 tečaka (kao i kod epsona dve susedne tačkice po horizontali ne smeju da budu ispisivane).

→



Brzi Gonzales

Ono što na možete da primetite ma koliko pažljivo posmatrali sliku, je fascinantna brzina rada štampača u standardnom modu specificirana brzina pisanja iznosi 180 karaktera u sekundi ali su naša merenja pokazala da pri potpuno popunjenim redovima ona dostiže preko 200 znakova Razlika u odnosu na 100 znakova kod epsona RX80 i 150 znakova kod FX60 je dakle lako primetna I porad znatnog ubrzanja glava štampača se praktično ne graje (probajte da pipnete glavu epsona pošto ispiše desetak strana) što bi trebalo da još produži vek koji prema specifikacijama iznosi 100 miliona znakova (oko 67000 strana) Da nastavimo sa poboljšanjima „kaga" je u normalnom modu primetno liša od „epsona a oba štampača osim toga poseduju mod u koma rada znatno sporija i znatno tiše

Glavna prednost „kaga' i „kanona" nad „epsonom se ipak zove malo drugačije NLQ mod U engleskoj literaturi se, naime često pominje termin „Letter Quality koji označava kvalitet otiska štampača sa lepezom i električnih pisaćih mašina, kvalitet koji je po njihovim standardima, jedini prihvatljiv za ozbiljna poslovna pisma NLQ je skraćenica od „Near Letter Quality što bi trebalo da označi otisak sličan pomenutom U kvalitet tog otiska ćete se lako uveriti kada bacite pogled na sliku 3 ma koliko se trudili nacate primatiti tačkastu strukturu slova Da stvar buda lepša konstruktori ovih štampača su se pobrinuli da u ovom modu predizajniraju karaktere tako da liče na slova IBM-ove pisaće mašina Da bi se taj kvalitat postigao bila je potrebna matrica 23'16 koja je dobijena uz pomoć trikova glava dve pute prelazi preko reda pri čemu je u drugom prolazu pomerena za pola tačke (ili možda za celu tačku?) naniže Obzirom da se u takvim prilikama ne može štampati u oba pravca I da preko svekog reda treba prelaziti dva puta brzina rada je smanjena na 30 znakova to je samo petnaestak procenata od uobičajenih 200 znakova ali je i dalje dvostruko brža od štampača sa lepezom!

Pošto je za definiciju jednog NLQ slova potrebno 23*16=368 bita ili 46 bajtova, konstruktori „kaga i „kanona su morali u njih da ugrade veliki ROM — umesto „epsonovih 6 K u „kanon su ugradena tri aproma 2764 što predstavlja ROM od 24 K Na štampi je ostavljeno i jedno prazno podnožje u koje možete da ugradite bilo dodatni osmokilobajtni aprom sa definicijama karaktera (u okviru uputstva za upotrebu nije, na žalost opisan tačan format tih aproma, varovatno zato što proizvodači nisu preveliki ljubitelji konkurencija) bilo RAM 6264 (8 K) koji zavisno od položaja jednog od 20 prekidača može da služi kao bafer ili kao prostor za definicija karaktera

Ćevapčići u Čačku

Kao što sta i očekivali štampači omogućavaju definisanja znakova, što je operacija neobično interesantna za ona koji se ne usuđuju da prčkaju po ROM-u sledeći naše savete iz Računara 3 Crtanja slova je spor posao koji zahteva pomalo umetnički dar, a pretvaranje crteža u brojke koje bi štampač razumeo mučenje čak i za najstrpljiviieg hakara Ukoliko vam se čini da je tako jednostavnije možela da sastavite nekakav editor za definicije karaktera na bejziku ili da se poslužite našim programom ispisanom na 6502 asembleru koji posebno objavljujemo u Biblioteci programa (str 37) interesovace vas jasno jedno DATA liste Predefinisali smo srednje i velike zagrade oznaku za funtu naopaku kosu crtu (backslesh) vertikalnu liniju i titlu kao manje-više napotrebne znakove koji se obično nalaze na lastaturama boljih kucnih računara u principu bi bilo jednostavno predefinisati neka od namačkih, švedskih francuskih danskih ili sličnih slova ali bi tada trebalo pisati printer drajver za tekst procesor koji vrši transformaciju koda

Ukoliko ste do sada koristili FX60 bilo kakav program za definisanja znakova koji poseduiete će raditi i na novom štampaču zahvaljujući potpunoj kompatibilnosti kodova Moracete jedino da se namučite ako želite da definišete NLQ karaktere za svako slovo treba pratvoriti 2*23=46 binarnih brojeva u dekadne pa otkucati rezultate Želeli smo da vam pomognamo i u ovome ali bezuspešno u uputstvu za upotrebu očigledno postoji neka greška zahvaljujuci kojoj štampač ignoriše pokušaje da se definišu NLQ slova jedan jedini primer koji je dat nije od naročite pomoci pošto ne radi ono što bi trebao da radi (u uputstvu za upotrebu cete uzgred budi rečano pronaci i mnogo drugih grešaka, ali ni jedna nije kritična kao ova jer možete da se poslužite uputstvom za epson FX80 koje je lakoda prošaraho greškama ali se ona srećom nalaze na različitim mestima)

Kontrolni kodovi

Osim raznih vrsta slova, „kaga' i kanon poseduju i gomilu kontrolnih kodova koji pružaju usluge odlično opisane engleskim terminom miscellaneous Tu spada ju horizontalna i vertikalna tabulacija (koliko god smo razmišljali nismo mogli da smislimo za šta bi se ove tabulacija koristile ukoliko je računar opremljan bilo kakvim taksi procesorom) poništavanja zadnjeg postelog slova pražnjenja čitavog bafera izbor veličina strana i aktiviranje automatskog preskakanja perforacija poslavljanja lava i desna margina zabrana dvosmernog štampanja (za ona koji vole da čekaju),

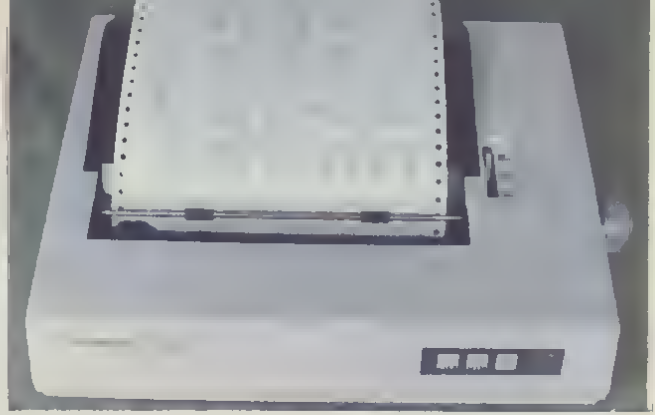
Doslo ja vrame da sa posvatimo glavnoj prednosti Kaga i Canona nad Epsonom: NLQ modu. U anglaskoj litaraturi sa, naima, casto pominja tarmin Lattar Quality koji oznacava kvalitat otiska Daisy Whaal stampaca i alektricnih pisacih masina, kvalitat koji ja, po njihovim standardima, jadini prihvatljiv za ozbiljna poslovna pisma. NLQ ja skracanica od Naar Lattar Quality sto bi trabalo da oznaci otisak slican pomanutom. U kvalitat tog otiska cata sa lako uvarita kada bacita poglad na sliku 3: ma koliko sa trudili, nacata primatiti tackastu strukturu slova. Da stvar buda lapsa, konstruktori ovih stampaca su sa pobrinuli da u ovom modu pradizajniraju karaktara tako da lica na slova IBM-ova pisaca masina. Da bi sa taj kvalitat postigao bila ja potrabna matrica 23*16 koja ja

isključivanje i uključivanje detektora nestanka papira i nerevno CHR$(7) za biper. Neke od ovih karakteristika možete da birate tako što ćete promeniti položaj jednog od mikroprekidača na štampanom kolu štampača, mada pre toga treba rasklopiti čitav uređaj što se pokazuje kao umereno komplikovan zahvat.

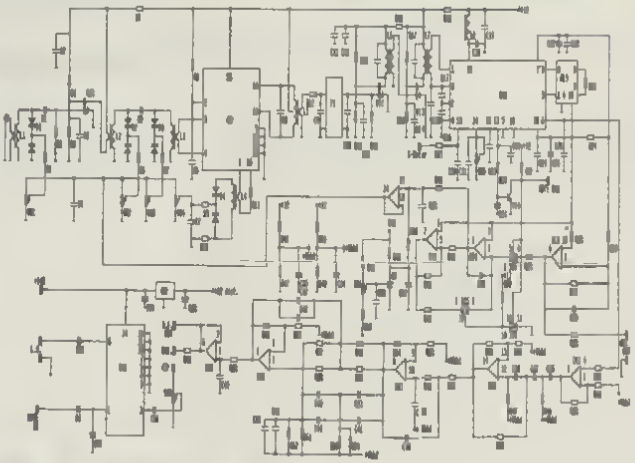
Grafičke mogućnosti „kege" i „kenone" su jednake epsonovim u normalnom modu može da se upravlja sa 60 tačaka po inču (=2.54 cm), u dvostrukoj gustini sa 120 a u četvorostrukoj gustini čak se 240! Razmak između linija može da se bira i iznosi n/72 ili n/216 inča (n sami birate). Svi programi za crtanje koji su radili na epsonu lepo će poslužiti i na „kagi" pa je tako nastao i neš crtež radi se o demonstraciji reda programa „Diegram" koji je pisan za BBC B onoga ko pogodi šta radi iscrtano kolo pozivamo da nam se javi.

Pre nego što završimo ovaj kratki prikaz trebalo bi da pomenemo i „sitnice" koje verovatno nestrpljivo očekujete od samog početka cene. Kaga KP810 se u Engleskoj može nabaviti za 290 funti a „Kanon PW1080A" za 320 što je i dalje manje od 360 funti koje treba izdvojiti za epson FX80. Ispleti li se onda kome da kupuje epsona? Možda, jer sa kod nas ovaj štampač mnogo koristi u raznim firmama penije teško (besplatno) doći do trake. Van ovih razloga Kaga KP810 i Kenon PW1080A predstavljaju izvenredne printere koje vam bez rezerve preporučujemo.

Dejan Ristanović



Novi standard u ekonomskoj klasi: Kaga KP 810





Časopis GALAKSIJA i ZAVOD ZA UDŽBENIKE I NASTAVNA SREDSTVA — Beograd raspisali su prošle godine STALNI OTVOREN KONKURS za izradu softverske i hardverske podrške za računare GALAKSIJA, SPEKTRUM i KOMODORE. Tokom protekle godine Zavod je od autore otkupio 13 kaseta sa više od 50 programa. Realizovan je i novi model računara GALAKSIJA+ sa visokom razolucijom trokanalnim zvukom i ekranskim editorom. Uz mnoge knjige koje su prihvaćene i štampane, prihvaćene je i ideje za proizvodnju robota u kitu. Zbog nesumnjivog uspeha koji je KONKURS imao i velikog interesa kod mladih stvaralaca iz cele zemlje obaveštavamo Vas da je i dalje

OTVOREN
STALNI KONKURS
za izradu softverske podrške za računare

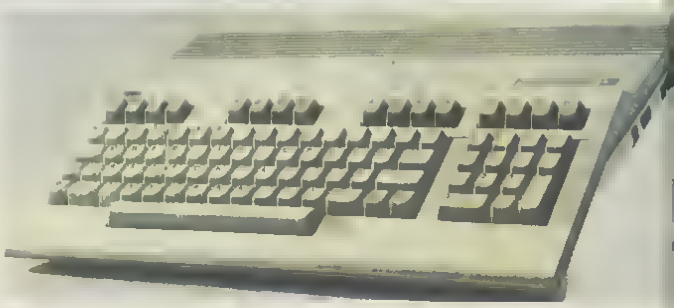
1 **GALAKSIJA** (4×6 i 8×6)
2 **GALAKSIJA+**
3 **SPEKTRUM** i
4 **KOMODORE 64**

Konkurs se odnosi na izradu
1 Sistemskih i uslužnih programa
2 Obrazovnih programa
3 Dalje usavršavanje hardvera i softvera za GALAKSIJU
4 Didaktičke igre
5 Elektronike u kitu (računari i roboti) i
6 Priručnici i knjige o računarima

Sve radove treba dostaviti na adresu Zavoda za udžbenike i nastavna sredstva, Beograd, Obilićev Venac 5/I telefon (011) 638-971 i (011) 638-405. Posle ocene Zavod sa autorima potpisuje ugovor o saradnji.

Računari u izlogu o „komodoru' sve najlepše Komodor 128

Pre nego što je komodor 128 zvanično i prikazan pričalo se da je C-128 samo prošireni C-64 Prvi utisci posle prikazivanja bili su da je to CP/M računar sa C-64 modom Istina je da je komodor 128 zaista kutija sa tri različita računara u sebi pa čemu je izgleda ipak najinteresantniji onaj — treći



Tri u jednom

Spolja sva tri računara dele istu odlično dizajniranu plastičnu kutiju sa 92 tastera Iznutra dele jedino RAM čipove i napajanje i — to je sve C 128 sadrži tri odvojene centralne jedinice dva odvojena bajzlika dva različita video čipa odvojena ROM memorije i čak različita memorijska mapa zavisno od toga u kome modu radi — C-64 modu C-128 modu sa 40 znakova C-128 modu sa 80 znakova i u CP/M modu sa 40 ili 80 znakova u redu Tu su i tri procesora 6510 sa 64 mod 8052 za C 128 mod (obe 6502 kompatibilna) i Z80A za CP/M mod Video čip sadrzi VIC II za 64 mod plus čip za C 128 i CP/M sa 80 karaktera Muzika je poverena jednom čipu za sva tri moda To je stari dobri SID čip čija mogucnosti su dobro poznate Programski jezici sadrze BASIC 2 0 za C-64 7 0 za C-128 i monitor za rad u mašinskom jeziku za C 128 i CP/M

Da li sve ovo zvuči zbunjujuca?

Zvuči ali — nije Svi modovi vam dopu šlaju da pralazila iz jednog u drugi bez resetovanja računara Kada uključile računar on prvo provarava da li se u disku nalazi CP/M disketa Ako jesta uključuja sa CP/M mod sa 40 ili 80 znakova (obično 80) Ako nije računar prvo provarava karlidž slot za C-64 Ako nada karlridž automalski sa uključuja u C-64 mod i startuja karlridž Ako ga na nade provarava da li je uključan C 128 karlridž U slučaju da ga ne nada proverava da li je pritisnul 40/80 DISPLAY taster i uključuje se u dogovarajuci C 128 mod

Jedna od najvecih dilama oko komodora 128 bilo je pitanje kompatibilnosti sa C-64 Zbog gorkih iskustava sa nekompatibilnom plus četvrtkom „Komodorovi ljudi su morali da se pobrinu da C 128 bude 100% kompatibilan sa C-64 Testovi pokazuju da C-128 radi sa svim poslojecim bejzik i mašinskim programima za C-64 Samo jedan program nije prošao test To je Turbo Disk — program koji ubrzava drajv 1541 Ti pula Program je lepo radio sa C 128 povezanim sa starim drajvom 1541 ali nija radio sa novim drajvom 1571 To i nije nekoo iznenadenja jer TurboDisk radi i sa memorijom disk-drajva i sa mamorijom računara Trabe uzeti u obzir da i novi drajv 1571 radi u tri moda kao 1571 kao stari drajv 1541 a moze da čita i original Osborne i Kaypro CP/M diskete Kada radi kao 1541 na disketu staje normalno 170K i brzina je ista kao kod 1541 ali kažu da se vrlo lako moze napraviti da drajv učitava i programe za C-64 Iz C 128 moda znači mnogo brže U osnovnom 1571 modu drajv radi pet pula brze i na jednu disketu staja oko 360K U CP/M modu drajv učitave 12 pula brže i na diskatu moža da stana 510K Diskdrajv koji je testiran još uvek je samo prototip i ljud iz Komodora tvrde da će raditi kada drajv buda završan, raditi i programi kao TurboDisk ili HypraLoad

Na upuštajući se previše u koncapcijska detalje prikazaćamo detaljnije novi komodorov bajzik nazvan 7 0 koji se koristi u C 128 modu To ja sudaći prama ocenama u stranim časopisima varovatno najmoćniji bejzik ikada ponuđan u parsonalnom računaru čak mnogo bolji i od IBM ili MSX bejzika Sadrzi sva komande kao bajzik 2 0 sve disk i fajl komanda iz bajzika 4 0 (C-8032 i superPet) skoro sva komanda za grafiku i muziku iz sajmonovog bajzika i supar grafike bejzika 3 5 (plus /4) i jos dosta loga

Bejzik 7 0 omogućava korišćenje grafike visoke razolucije sprajtova i muzike bez starih PEEK i POKE komandi ili mašinskog jazika Pokralanje sprajtova sa odvija prako interapta lako da nakoliko bejzik komandi mogu istovramano da pokrece osam sprajtova za vrame dok program radi nešto drugo pa čak i kada je program prekinut Za definisanje sprajtova mozale nacrtat nešto u visokoj rezoluciji pa to prabacili u sprajt ili koristili ugradani sprajt editor Korišćenje SID muzičkog čipa je takode veoma pojednostavljeno tako da možete svirati ili praviti zvučne afekta sa svega nekoliko bejzik komandi

Zbog vrlo obimnog i razradenog bajzika opisaćamo samo vaznije interesantnija naredba iz onih oblasti sa kojima je komodor do sada uvek oskuđevao

Računar za sve. Zahvaljujući kompatibilnosti se „sedamsetčetvorkom" na „komoderu 128 sa mogu igrati izuzetno atraktivne igre dok CP/M mod otvara vrata u „akademsku biblioteku poslovnog softvera sa preko 10000 ozbiljnih naslova

Disk

Disk i fajl komande sadrze DLOAD i DSAVE (za snimanje na disk i sa diska bez dodavanja ili ime programa) OVERIFY (za ver fikaciju programa na disku) CATALOG i DIRECTORY (za prikazivanje spiska fajlova na diskati bez brisanja bejzik programa) COPY (kopiranja programa korišćenjem duplog diska) BACKUP (kopiranje cale diskete korišćenjem duplog diska) COLLECT (reorganizacija blokova na diskeli) CONCAR (za kombinovanje dva programa) HEADER (za formatiranje diskete) RENAME (promena imena programa) SCRATCH (brisanje programa) DOPEN i DCLOSE (otvaranje i zatvaranja kanala) DCLEAR (zatvaranje svih disk kanala) DS i DS$ (za čitanje kanala grašaka) BLOAD i BSAVE (za snimanja mašinskog programa ili dala mamorije) i BOOT (za automatsko učitavanja i startovanje programa)

Nake komande su vrlo slična kao za C-64 i 1541 drajv na primer COLLECT je isto što i OPEN 1 8 15 VO CLOSE1 Komanda se takode mogu koristili kao D-SHIFT-L za DLOAD ili čak SHIFT RUN STOP za automatsko učitavanja i startovanje programa sa diskete Takode neki funkcionalni taster su program rani lako da izvršavaju odredene komanda kao na primer DIRECTORY

Sprajtovi

Komande za rad sa sprajtovima ne zamenjuju samo stare POKE komande već pružaju mnogo vace mogucnosti U demo programu koji ide uz računar bajzik 7 0 pokrece sprajtove dovoljno brzo da možete praviti odlične akcione igre bez mašinskog jezika

sprdat

U direktnom režimu komanda aktivira ugradeni sprajt editor Komande uz sprajt editor omogućavaju vam da sprajtove brišale ispra

Jedan od najmoćnijih trendova u današnjoj softverskoj industriji je integrisani softver softv x paket koji sadrži dva tri ili više različitih programa u jednom kao na primer, LOTUS 1-2-3 Sada nam „Komodor" prvi put predstavlja integrisani hardver tri računara u jednoj kutiji i to sve za manje 300 dolara! Ova kutija sadrži standardni „komodor" sa 64 K memorije novi C-128 mod sa novim bejzikom 7.0 i CP/M mod baziran na procesoru Z80 sa svojih 10000 profesionalnih programa Korišćenjem RAM diskova memorija se može proširiti do 512 K Kada se tome doda grafika visoke rezolucija 640×200 osam sprajtova trokanalni generator tona i, neki kažu, najmoćniji bejzik ikada ponuđen u nekom kućnom računaru postaje jasno da oni koji vole „komodor" ovoga puta dobijaju premiju.

Rečnik bejzika 7.0

automatska numeracija reda
otvara sekvencijalnu datoteku
kopira celu disketu
poziva blokove memorije za PEEK POKE SYS
postavlja više bejzik linija u jedan blok
učitava i startuje CP/M sa diskete
crta pravougaonike
snima deo memorije
određuje tip dodira sprajtova
lista sadržaj diskete
prebacuje leksi u visoku rezoluciju
crta krugove elipse i sve poligone
briše otvorene datoteke i reorganizuje disk
određuje dodire sprajtova
boje leksika i grafike
povezuje dve sekvencijalne datoteke u jednu
kopira čitavu disketu sa diska
zatvara sve kanale prema disku
zatvara pojedinačan kanal prema disku
pretvara heksadecimalne brojeve u decimalne
briše određeni deo programa
izdaje specijalne programa sa diska
učitava sa diska
otvara kanal prema disku
struktura „pallje
crta lučke linije
snima na disk
čita kanal za greške sa diska
daje leksi greške sa diska
proverava program sa diska
prilagođuje programsku liniju u kojoj je greška
deo IF THEN naredbe
dalinišeki vu za sintesajzer
daje kod greške
daje leksi greške
izlazak iz DO LOOP petlje
prelazak na rad sa tastaturom od 2MHz
uzima podatke iz memorije RAM FLOPPY
postavlja parametre za SID čip
čeka na pritisak nekog tastera
uključuje C-64 mod
bira grafički mod
prilagođava sliku u visoku rezoluciju
formatira disketu
lista grafike

pretvara dec u heksadecimalne
određuje poz stringa u drugom stringu
lapliuje džojstik portove
dalinišeka funkcionalna tastatura
pozicionira grafički kursor
poziva ugrađeni mašinski monitor
pomera spajt
ispunjava prostor u visokoj rezoluciji
ispituje svaštono pero
svira tonova iz nekog stringa
daje adresu veličine u memoriji
ispituje padl kontrolera
omogućava dalinisanje karaktere
dalinišeke karaktere za PRINT USING
ispituje boju za taster grafiku
formira relativna datoteka
menja ime fajlu na disku
vrši planumaracju programskih linija
postavlja brojač za DATA na određenu liniju
deje broj grafičkog moda
deja vrednosti A X Y i status flege
daje kod multikolor moda za spajtove
deje poziciju i brzinu spajta
deja sva spajt parametre
deje sva prozori parametra
briše grafički ekran za teksi
briše fajl sa diska
prilagođava visoke rezolucije u string
zadržava izvršavanja programa za zadato vrama
prilagođava 1MHz taster
određuje frekvenciju i trajanje tona
boja spajta
poziva ugrađeni spajt editor
postavlja parametra za spajtove
prilagođava spajt u string i obrnuto
prevod DATA podatka u memoriju
viši razmenu podataka između dva bloka od 64 K
određuje brzinu za PLAY
isključuje TRACE
uključuje TRACE
postavlja uslove za Do (UNTIL) LOOP petlje
jačina zvuka za SOUND
DO (WHILE) LOOP
određuje debljinu linija za crtanje
određuja veličinu prozora
ekskluzivno ILI

U spisku se ne nalaze komande bejzika 2.0 za C-64 VIC 20 već samo nova komanda bejzika 7.0

vlajte ili im menjate boju i to u običnom ili multi kolor modu. Kada završite definisanje spajtova vratite se u bejzik U bejziku 7.0 na morate da brinete gde ćete smestiti spajtove u memoriji. Računar će ih odvajati memoriju počev od lokacije 3584 (E00 hex.) U ostalom sa 128K RAM memorije sigurno nećete imati problema sa nekoliko spajtova.

SPRITE ,an/aff foreground priority, x y mode

Postavlja odredene parametre kao što su uključivanje i isključivanje spajtova prioritet postavlja spajt na određeno mesto i određuje singl ili multi-kolor mod

SPRSAV sprite string
SPRISAV string sprite

Prebacuje spajt u string i obrnuto. Na primer možete prvo nacrtati nešto u visokoj rezoluciji zatim to pretvoriti u string komendom SSHAPE i na kraju string pretvoriti u spajt. Možete lakoda šetati spajt po ekranu i kada požalite da vam slajno slopi na jednom mestu jednostavno ga pretvorite u grafiku komandama SPRSAV i GSHAPE. Tako možete poništi da vam cele slike visoke rezolucije dođe iz raznih pravaca i sastavi se na ekranu.

MOVSPR sprite#X, Y

Ova naredba se zove apsolutno pomerenja spajtova i ona pomera određeni spajt na koordinata određena sa X Y. Slična komanda kao standardno POKE kod C-64

MOVSPR sprite#,+/−X,+/−Y

Ova naredba se zove relativno pomeranja spajtova i koristili se kada ne znate gde vam se spajt nalazi na ekranu. Određeni spajt se pomera za X i Y u odnosu na stare koordinate. Na primer ako želite da pomerite spajt pet za sedam mesta ulevo i deset masta nadole kuca ćete MOVSPR 5 −7 +10

MOVSPR sprite#, angle# speed

Pomera spajt u određenom pravcu određenom brzinom. Ovo je urađeno prilično mašinskog iniciranja. Iako da se spajtovo kraću po ekranu za vreme dok sa bejzik programi i dalje izvršava. Kada spajt izađe se ekrana pojavljuje sa automatski na drugoj strani ekrana. Na primer možete sa pet bejzik komandi naprevili da vam se spajtovi kreću po ekranu u nekoj igri lipa osvajače dok vam program ispituje džojstik i pomeri vaš brod po ekranu. Parametri angle određuje pravac kretanja spajtova 0=gore 90 desno 180 dole speed brzinu pomeranje

COLLISION type line#

Ova komanda tag-iruje dodir spajta sa drugim spajtom dodir sa pozadinom ili sa lajt panom (ovo zavisi od parametra type). Parametar

→

Za one koji znaju malo više

linija# označava broj programske linije podprograma na koji program treba da pređe ako dođe do nekog dod ra Potprogram mora da se završi va sa RETURN kao i svak potprogram

BUMP (type)

Ispituje dodira sprajtova i vraća ih kao vrednost Pomoću njega možemo detektovati i dodire koji su se dogodili izvan dela vidljivog na ekranu BUMP 0 registruje dodire između sprajtova a B

Grafika

GRAPHIC mode#, clear, window

Prebacuje ekran u mod određen parametrom MODE# Mod 0 je mod za laksi sa 40 znakova u redu 1 je viseka rezolucija 2 je visoka rezolucija sa prozorima 3 je multi-kolor mod 4 je visebojni mod sa prozor ma i 5 je mod za laksi 80 znakova u radu (samo RGB) Prozori su sli kao kod apli i ajori računara ; predstavljaju nakoliko linija teksla na ekranu visoke rezolucije Počinju od linije 18 ali se mogu pomerati pomoću parame tra window Parametar clear određuje da li će ekran biti obrisan 0 - da 1 - ne

COLOR source#, color#

Postavlja boja na ekranu Color# parametar označava boju od 0 do 16 a parametar source određuje ekran 0 za 40 znakova pozadinu 1 za pozadinu u visokoj rezoluciji 3 za boju okvira (bordera) 5 za boju karaktera i 6 za boju pozadina kod ekrana sa 80 znakova

BOX source#, Xcentra, Ycentre, Xradius, Yradius, arc angle1, arc angle 2, angle increment

Crta na ekranu visoke rezolucije krugova slične pelouglova osmouglove i sve ostala poligona Source# ja boja tačaka X i Y centra su koordinate centra X Y radijus su X i Y koordinata radiusa arc angla 1 početni ugao u stepenima (najbolje 0) arc angla2 je krajnji ugao u stepenima (360) angle je ugao rotacije (0) a incrament određuje ugao između segmenata (2) Na primer CIRCLE 1 180 100 65 50 crta zeleni krug CIRCLE 1 180 100 65 10 zelenu elipsu CIRCLE 60 40 20 1 45 osmougao a CIRCLE 260 40 20 80 crta dijamant

DRAWN source#, X1, Y1 to X2, Y2 to

Crta tačke ili linija na ekranu Source označava boju tačaka X1 i Y1 su početne koordinate linija X2 i Y2 su krajnja tačka Na primer DRAW 1 150 100 TO 250 100 crta horizontalnu liniju a DRAW 1 10 10 TO 150 100 TO 250 200 TO 10 10 crta trougao Ako vam tačka zadu izvan ekrana odnosno ako su vam koordinata racimo negativna na brinila C 128 će povući liniju do neke zamišljene tačke izvan ekrana i neće vam prijaviti nikakvu grešku

LOCATE X, Y

Postavlja nevidljivi grafički kursor na koordinate X Y kao početni položaj za dalje crtanje (DRAW TO A B)

PAINT source#, Xstart, Ystart, mode

Popunjava deo ekrana bojom određenom parametrom source# X i Y označavaju početna tačke a mode određuje kako da se izvrši popunjavanja (0 = popuni deo označen sa source# 1 popuni deo označen tačkuma na ekranu)

SSHAPE string, corner 1, corner 2

Pravi ara određeni pravougaonik nacrtan na ekranu visoke rezolucija u string Površina ovog dela može biti do 255 bajtova koliki je jedan bajtik string SSHAPE je vrlo slična komanda kao GET kod IBM bajtika

GSHAPE string, ceranr 1, cornar 2, mode

Pravi ara podatka iz odradanog stringa u tačka na ekranu Cornar 1 i 2 odrediju koordinate na ekranu visoka rezolucije a mode označava način kako će tačke biti postavljane na ekran 0 pravougaonik će biti smešlan onako kakav i jesle 1 invertuja pravougaonik 2 radi logičko OR (ILI) između tačaka na ekranu i tačaka iz stringa, 3 radi logičko AND (I) 4 EOR (ekskluzivno ILI) Komanda slična kao PUT kod IBM bajtika

Zvuk

SOUND voice, freq, dur, sweep, min, step, wave, width

Postavlja zvučna parametra Vo ca označava glas (1—3) freq frekvenciju 0—65535 a dur 0—32767 in fys (jiffy = 1/60 sec) Ostali parametri podešavaju zvuk do poslednjih stepica

PLAY O oct, T tune, U vol, V voice, X filter, notes

Svira jednu ili više nota koristeći odabranu oktavu anvelopu glasnoću glas i filter Oct označava oktavu 1—8 Juna označava anvelopu 0—9 vol jačinu 1—8 voica glas 1—3 filtar (0 = uključen 1 = isključen) Posebno je interesantan tyna parametar koji označava instrumenta 0 piano 1 akorde 2 kalliope 3 bubnja 4 flauta 5 gitara 6 klavikord 7 organa 8 trube 9 kalioion

ENVELOPE#, attack, decay, sus, rel, wave, width

Ovom komandom možna sa predefinisati neki instrumenti odradan parametarom Juna

FILTER freq, lopass, bendapass, hipass, res

Ova komanda uključuje i isključuja filtere zadata filtar parametrom kod komande PLAY Posloja čalira vrsta filtera koja možela koristiti za pisanja muzika ili pravljanja različitih zvučnih efekata

Pored komandi za sprajtove grafiku i muziku komodora 128 sadrži još dosta drugih komandi kao što su komanda za prozora za uključivanja i isključivanja blokova od 64K brisanja dalova bejzik programa postavljanja funkcionalnih tastera pronalaženja grešaka u bejzik programima Takoda posloja i komande za uključivanja mašinskog monitora zalim sa rad sa parijama bez komandi FOR-NEXT i komanda za strukturirano programiranja

Bajzik 7.0 predstavlja izuzalan domet u razvoju programskog jezika za programiranja na visokom nivou Potpuno isključuja potrebu sa najnižim komandama tipa POKE i PEEK Pored ovog obimnog bejzika Komodora 128 ima ugradani i odličan mašinski monitor za ave ona kojima ni ovoliki bejzik nija dovoljan i koji žela da iz svojih mašina izvuku — još više

Vladimir Krstonošić

Računari u razgovoru

stiven džobs — programer koji je hteo da promeni svet

Nigde, valjda, toliko često ne padaju krunisane glave kao u industriji kućnih računara. Pre nekoliko meseci na spisku penzionisanih kraljeva našao se i Stiven Džobs, projektant čuvene „jabuke" i osnivač kompanije „Apple". Dok je boravio u Japanu u „misiji dobre volje", Džobsa je u dinastija-stilu svrgao sa prestola Džon Skuli, čovek koji zna sve o tržištu i gotovo ništa o računarima. Dodeljena mu je utešna kruna, utešna kancelarija i utešna sekretarica. Ovaj razgovor sa Džobsom, objavljen u časopisu „Playboy", voden je pre prevrata, krajem prošle godine — u trenutku kada se mislilo da je „Apple" definitivno lepljivo iz teškoća u koje su ga uvelili računari „liza" i „epl III". Govoreći o sadašnjosti i, još više, o budućnosti kućnih računara, Džobs otkrivе suptilne zakonitosti i mehanizme u ovoj mladoj industriji, u kojoj su jabuke, izgleda, znatno kiselije nego što to ne previ pogled izgleda

Ako se za koga može reći da predstavlja duh poduzetničke generacije onda je čovjek kojeg treba tražiti karizmatički suosnivač i predsjednik Apple Computer Inc. Stiven Džobs. On je transformisao malo preduzeće koje je nastalo u garaži u Los Altosu, Kalifornija u revolucionarnu kompaniju od milijardu dolara — onu koja je dostigla nivo Fortuna 500 (lista od 500 najvećih kompanija koju objavljuje časopis Fortune) za samo pet godina brže nego i jedna druga kompanija u istoriji. I ono što najviše pogađa u cijeloj stvari je to da je ta osoba stara samo 25 godina.

Džobsova kompanija je uvela personalne računare u američke kuće i radna prostorija. Prije osnivanja Apple a 1976. slika koju su ljudi imali o računarima je ona o mašinama iz naučnofantastičnih filmova koja su davala svjetlosne i zvučne signale ili pak o onim velikim mašinama koje u tišini zlosutno čama iza vrata velikih korporacija ili vladinih agencija. Ali sa razvojem tranzistora a zatim i mikroprocesorskog čipa postalo je moguće minijaturizirati tehnologiju računara i učiniti ih dostupnim pojedinačnim korisnicima. Do sredine sedamdesetih godina početni računarski kit uglavnom interesantan samo hobistima mogao se nabaviti za oko 375 dolara plus dodatni pribor.

U dolini južno od San Franciska već poznatoj po koncentraciji elektronskih firmi i mladih kompanija koja su tek startala sa radom dve prijatelja koja su dijalila sklonosti za napravama i elektroniku dala su sa na posao da stvore vlastiti računar. Džoba tada star 21 godinu usvojanj sin mašinista praučeo je posao dizajniranja video igara kod Atarija nakon što je napustio REED koledž dok je Vozniak (Stephen Wozniak) 26 radio kao inžinjar kod Hewlett Packard a jedna bo najvećih kompanija u regiji zvanoj Silicijanska dolina. U svoja slobodno vrijema prijatelji su projektovali i izgradili improvizirani računar u stvari samo ploču sa elektronskim krugovima koju su hirovito nazvali epla I. On nija radio mnogo stvari ali kada su sa suočili sa listom poruđžbina bo 50 komada Džobau je svanulo da se možda stvara rasluđa. Tržišta personalnih računara.

Vazniakov interes je bio primarno tehnički Džobs se bacio na to da učini računar pristupačnim ljud ma. Zajedno su dodali tastaturu i memoriju (na „epla I" a Vozniak je razvio disk drajv i dodao video terminal. Džobs je angažirao eksperte da osiguraju sigurno napajanja i ljapo kućišta i tako je rođen apl II — zajedno sa cijalom industrijom.

Uspon Apple a je bio meteorski. Od prodaje od 200 000 dolara te prve godine u Džobsovoj garaži (verzija Linkolnovog uspjeha ali sad u Silicijumskoj dolini) kompanija je narasle u gigantsku firmu sa prihodom od 1.4 milijarde dolara u 1984 godini. Njeni osnivači su postali multimilionari i narodni heroji. Vozniak koji se stvarno povukao iz Apple a 1979 da bi se vratio na koledž i posvetio sponzorstvu nad muzičkim festivalima imao je malo posla sam svog kreativnog doprinosa tehnologiji. Džobs je osiao da vodi kompaniju da vidi da 70% kućnih i školskih računara nosi znak Apple a da se bori protiv snaga u Appla u koja žala da ga dalronizraju i najviše od svega da sa bori sa IBM om divom od 40 milijardi dolara kako ga inače zovu od kada je odlučio da uđa u posao sa personalnim računarima.

Sa procijenjanim čialim imelkom od 450 000 000 dolara, uglavnom u akcijama Apples. Džobs je dalako najmlađa osoba na Forbesovoj listi najbogatijih Amerikanaca već nekoliko posljednjih godina (Vrijedno ja napomanuti da je od 100 Amerikanaca koja imanuja Forbes Džoba jedan od sedmorica koji su svoj imalak zaradili sami.) Nedavno sa padom vrijednosti akcija Apple-a, za vrijama laškoća u 1983. on je izgubio skoro četvrt milijarda dolara na papiru. Iako da mu se čisi imatak danas procijenjuje na 200 000 000 dolara.

Ali iraba vjarovati Džobsu da mu novac ne znači mnogo pogotovo jar ga na troši praljerano i zaista nama mnogo vramana za društvani život. On ima misiju — propovjeda spasenja uz pomoc personalnog računara po mogućnosti Apple-ove proizvodnja. On je vrlo angažirani prodavac i nikad na propušta priliku da proda svoj proizvod rjećito opisujući vrijama kada će računari biti u tako širokoj upotrabi kao kuhinjski aparati i tako revolucionarni po svom uticaju kao telefon ili motor sa unutrašnjim sagorjavanjem. Činjanica ja da iranuino ima više od 2 000 000 apl računara i procjanjuje sa 16000 programa — u učionicama dnavnim sobama u pradgradu Jar mama slanicama za lansiranja rakata. Ta malim i velikim praduzaćima diljam Amerike.

Stvarajući široko Izršita računara Apple je stvorio i uslove za konkurenciju i mnoštvo kompanija je ušlo u borbu za osvajanja Izršita na kome je Apple dominirao do 1977. do 1982. Ali ni jedan drugi proizvod nija bio tako uspješan kao IBM-ov PC koji je ubrzo zauzeo 28% Izršita. usposljavajući novi standard. U usiovima smanjivanja svog Izršita Apple je pradslavio dva nova računara. Ijau i apie III ali prijam nija bio oduševljen. Sredinom 1983 analitičari su sa glasno pitali da li će Apola preživjeti.

Tokom trvenja unutar kompanija Džobs je preuzeo odjeljanje Apple-a koja je gradilo kompletno nov računar za koji je on mislio da je Apple-ove najbolje i posljednje šansa. To nije bile uskogrudnosti kaže on jer će se propali, IBM bi ostao da dominira i upropastio bi industriju. Nakon tri godine lansiranja je mekintoš (Mecintosh) uz reklamnu kampenju od 20 000 000 dolara. Objavljen kao „računer za nas ostale" razglašen je kao gigantski korak u



(Ne) slažne braća. Stiven Džoba novi predsednik Apple a Džon Skuli i Stiven

proizvodnji računara koji su laki za korišenja. Sa akranom bijelim kao papir. malim slićicama koja predstavljaju programske opcija i mišem. mak je ajurno najmanja zastrašujući računar koji je ikad napravljen. Bio je lakodar kritikovan da je isuviše igračka i nepodesan za ozbiljnu poslovnu upotrabu. Uprkos žaslimi ih argumanala Apola je vrijedno proizvodio 40 000 makintoša mjesačno a ova godina ima namjaru da proizvodnju udvostruči.

Zavisno od loga s kim razgovarate, za jedne ja Džobs vizionar koji ja promijanio svijat na bolje a za druga oportunista čija su vješina u merkelingu dovele do nevjarovatnog komercijalnog uspjeha. U fermerkame i iznošenim tenisaljalkama on vodi kompaniju koja sa ponosi time da ima u sabi antuzijazam šezdesetih i poslovnost osamdesetih. Iako da mu sa dive i da ga se boje. On je tu zbog koga radim 20 sati dnavno kaže jedan inženjer III. keko Michael Moritz izvještava u Malom kraljavstvu. Džobsove tvrdoglavosti — da kuja hvalospjeve jeden dan e

→

izvrgava ruglu drugi dan — umalo je dovala do raspada firma koji je radio na mekintošu. On je neposlojanog predsjednika Pepsi Cola Džona Skulija (John Sculley) pitao da preuzme administrativnu upravu Apple a njegova. Hoces li da naslavis da prodajes djeci zasladanu vodu kad možeš da mijenjaš svijet? Skuli je prihvatio ponudu.

Ovaj intervju ja bio jedan od malo onih u mom životu kada mi se stalno dogadalo da budem prelijerano obučen. Čuo sam o neformalnosti koja vlada u Apple u ali ipak ja sam intervjuisao glavu kompanije od milijardu dolara pa sam na prvi sastanak došao sa kravatom. Naravno, kad sam sreo Džobsa u njegovoj kancelariji u Kapertinu, Kalifornija on je nosio flanelsku kosulju i farmerke. Jos se uvijek nisam osjecao deplasirano — dok nisam sreo Džona Skulija novog predsjednika Apple a. On je nosio majicu sa kratkim rukavima.

Kancelarije u Apple u definitivno ne nalikuju ostalim radnim mjestima. Svuda naokolo video igre, slotovi za slot tenis su u stalnoj upotrebi, zvučnici trešte od muzike Rolling Stones a do Windham Hill džeza. Sale za sastanke se zovu po Da Vinčiju i Pikasu a ir žideri u snek prostorija ma su prepuni svježim sokovima od mrkve, jabuke i narandže. (Samo mek tim potroši 100 000 dolara godišnje na sokove).

Nadugo sam razgovarao sa Džobsom i na poslu i na njegova jedina dva odmora tokom godine u Aspenu i banji Sonoma gdje je trebalo da se opusti. Nesposoban da uspori svoju misiju da čini nješ Apple-a sa izuzetnom ošitrinom je govorio o svom stavu sa IBM om da bi onda podcrtao svoj entuzijazam za neku ideju rijecima. Cisto! ili Nevjerovatno grand ozno!

• Preživjeli smo 1984. a računari nisu osvojili svijet mada neki ljudi u to tesko vjeruju. Ako postoji pojedinac koga treba bilo opljuvati bilo hvaliti za prodaju računara u 29 togodišnji otac kompjuterske revolucije sa prvooptužen. To vas je takoder učinilo bogatim preko svih snova — vase akcije su u jednom momentu vrijedile skoro pola milijarde dolara, zar ne?

— Ja sam u stvari izgubio 250 000 000 dolara u jednoj godini kad su akcija pale (smije se)

• Možete da se na to smijete?

— Necu dozvoliti da mi to upropasti zivot. Zar to nije na neki način i smijesno? Znate moja osnovna reakcija na to pitanje novca je da je to smijesno. Sva ta paznja koja se posvecuje lomaljer teško da je to najvece saznanje ili najvrijedniji ja stvar koja mi se desila u posljednjih deset godina. Osecam se ponekad starim kad govorim na univerzitetu i uvidim da najvise strahopostovanja studentima uliva činjenica da sam milioner.

• Interesantno je da je kompjutersko poduc je učinilo milionere od

Mladih momaka znam.

• Htjeli smo da kažemo od ljudi kao što ste vi i Stiv Voznjak koji su radili u garazi prije tek deset godina. Šta je u stvari ta revolucija koju ste započeli vas dvojica?

— Mi živimo nad opelom petrohemijskoj revolucij od prije 100 godina. Petrohemijska revolucija nam je dala jeftinu energiju — jeftinu mehanicku energiju u ovom slučaju. To je promijenilo tekstura društva u raznim vidovima. **Ova revolucija informatička revolucija je revolucija jeftine energije takoder, ali energije druge vrste jeftine intelektualne energije.** Ona je još vrlo sirova danas a ipak racunar mekintoš troši da bi radio manja energije nego sijalica od 100 vata a može da vam uštedi sate dnevno. Šta ce biti u stanju da uradi za deset ili dvadeset godina od danas? Ova revolucija će zasjeniti petrohemijsku revoluciju. Mi smo u prvoj liniji fronta.

*• Kako bi bilo da nam date neki **konkretan** razlog zašto kupiti racunar danas? Jedan ruko vodilac iz vaše branse je nedavno rekao: Dali smo ljudima računare ali im nismo pokazali šta da rade sa njima. Ja mogu bilansirati svoju cekovnu knjizicu ručno brže nego pomocu računara. Zašto bi neka osoba trebala kupi racunar?*

Postoje različiti odgovor za razna osobe. U poslu na to pitanje je lako odgovoriti: vi stvarno možete dokumente pripremiti mnogo brže sa vecim nivoom kvaliteta i možete uradili mnogo stvari da povecate produktivnost biroa. Racunar oslobadja ljude od mnogo prostog posla. Sem toga dajeta im alat koji ih ohrabruje da budu kreativni. Sjetite se racunari su alatke. Alat nam pomaže da bolje obavimo svoj posao. U obrazovanju računari su prva stvar koja se pojavila poslije knjige koja ostaje sa vama u interakciji do u beskraj. Sokratovo podučavanje se više ne može obezbijediti a računari imaju stvarni potencijal da predstavljaju prvi probor u obrazovnom procesu kada se koristie u vezi sa prosvjetiteljenim učiteljima. Mi smo već u mnogim školama.

• To su argumenti za uvodenje računara u posao i škole ali šta kažete za kuce?

Za sada to je tek izišle u zacetku. Primarni razlog da kupite računar za svoju kucu je taj da želite da radite neki posao kod kuce ili želite da kor stite neki obrazovni softver za sebe ili svoju djecu. Ako kupovinu računara ne možete opravdati jednim od ova dva razloga jedini drugi razlog je taj da želite da budale računarski pismeni. Znate da se nešto dešava ali ne znate tacno sta je to i sloga želite da naučite. Ovo ce se izmijeniti. Računari ce biti neophodni u vecini kuca.

„Novac nije najvece saznanje niti najvrednije stvar koje mi se desile u posljednjih deset godine"

• Šta ce se promijeniti?

— Najizazovniji razlog za vecinu ljudi da kupe racunar za kucu ce biti veza na nacionalnu komunikacijsku mrežu. Upravo se nalazimo u početnom stadiju onoga što ce biti istinski zavidan napredak za vecinu ljudi — isto toliki koliki je bio telefon.

• O kakvoj specifično vrsti napretka govorite?

— Mogu samo početi da spekulišem. Dosta toga vidimo u našoj industriji. Ne znate tacno cime ce to rezultirati ali znate da je to nešto veoma vaiiko i veoma dobro.

• Dakle zar vi za sad ne trazite od kupaca kucnih računara da investiraju 3000 dolara u nesto što je u biti akt vjerovanja?

— U buducnosti to nece biti akt vjerovanja. Teškoce sa kojima se sada srecemo su te da ljudi traze da mi govorimo o detaljima o kojima ne možemo govoni. Da je prije sto godina neko upitao Bela: Šta cete moci uraditi sa telefonom? on ne bi bio u stanju da mi objasni u kojim sve vidovima ce telefon uticati na svijet. On nije znao da ce ljudi koristiti telefon da saznaju koji film gra veceras da kupe naku robu ili da porazgovaraju sa rodbinom na drugoj strani globusa. Ali prisjetite se da je prvi javni telegraf bio inauguriran 1844. To je bio zaciudujuci probor u komunikacijama. Mogli ste tada poslati poruku iz Njujorka u San Francisko u toku jednog popodneva. Ljudi su govorili o tome da treba staviti po jedan telegraf na svaki radni stol u Americ. da bi se povecala produktivnost. Ali to nije radilo. Bilo je potrebno da ljudi nauce cijeli niz cudnih znakova Morzeovu abecedu tackica i crtica da bi koristili telegraf. Trebalo je oko 40 sati da sa to nauci. Vecina ljud ne bi nikad to naucila. Ali srecom 1870. Bel je patentirao telefon. On je u osnovi radio isto sto i telegraf ali su ljudi vec znal kako da ga koriste. Takode sem što je dozvolio komunikaciju i rijecima dozvolio je i mogucnosti da pjevala.

• Šta sad to znaci?

— Omogucio je da date rijecima intonaciju sa znacenjem koje ja iznad cisto lingvistickog. Mi smo u isto j situaciji danas. Neki ljudi misle da na svaki radni stol u Americi treba staviti IBM PC da bi se povecala produktivnost. To nace funkcionirali. Specijalne carolije koja ovog puta treba da naucile su crtica q zs i slicne stvari. Upustivo za WordStar najpopularniji program za obradu teksta debelo je 400 stranica. Da biste napisali roman morate procitati roman — i to roman koji je kao mister ja za mnoge ljude. Oni nece željeti da nauce crtica q sz nimalo vise nego što su željeli da nauce Morzeovu abecedu. To je ono što cini mekintoša. To je prvi telefon naše industrije. A osim toga ja mene je najvažnije da mekintos dovoljava da pjevate na isti nacin na koji je to cinio telefon. Ne komunicirate jednostavno rijecima — imate specijalne stilove šiampanja i mogucnosti da crtate i dodate slike da biste se bolje izrazili.

• Vecina kompjutera koristi tastaturu za unosenje instrukcija a mekintos je zamjenjuje necim što se naziva miš malom kutijom koja se kotrlja po radnom stolu i time vodi kursor po ekranu. To je velika promjena za ljude koji su navikli na koristenje tastature. Zašto mis?

— Ako želim da vam kažem da imate mrlju na košulji necu to raci rijecima. Ima jedna mrža na vašoj košulji 14 cm niže od kragne i 7i centimetara lijevo od dugmeta. Ako ima mrlje – ovde (on pokazuje) — ja cu vam to pokazati prstom. Pokazivanja prstom je metafora koju svi mi znamo. Izradili smo mnogo studija i ispitivanja o tome i ustanovljeno je da se sve vrste funkcija izvode ne samo lakse nego mnogo brže uz pomoc misa.

• Koliko je trebalo vremena da se razvije mekintoš?

— Za izradu samog racunara trebalo nam je više od dvije godine. Na tehnologijama koje su primijenjene radili smo godinama prije toga. Rad na mekintošu bio je najljepše iskustvo mog zivota. Skoro svako ko je na njemu radio reci ce to isto. Na kraju niko od nas nije htio da ga pusti. Kao da smo znali da kad ga jednom ispustimo iz ruku više nikad nece biti naš. **Kad smo ga konacno prikazali na sastanku akcionara avi u auditorijumu su ustali i pozdravili ga petominutnim ovecijama. Svi su pecali da plaču.**

• Bili smo upozorani na vas prije nego što je ovaj intervju započeo neko je rekao da „cemo biti zasuh lavinom superlativa.

— Mi smo samo zaneseni onim što radimo.

• Ali baš imajuci u vidu taj aniuzijazam reklamnu kampanju od više miliona dolara i vašu vlastitu sposobnost da pridobijete publicitet štampe, kako da potrošac zna šta se krije iza svega je buke?

— Reklamne kampanje su neophodne radi konkurencije — reklame IBM-a su svuda. Ali dobro propaganda obrazuje ljude i to je sve. U ovom poslu ne možete zavarati ljude. Proizvodi govore sami za sebe.

• Pored nekih ponovljenih kritika — da je mis neefikasan da je ekran mekintosa samo crno bijeli najozbiljnija opljuvba je da epleprecjenjuje svoje proizvoda. Bi li vam smetalo da odgovorite na neku ili sve ove optužbe?

„Ova revolucija, informatička revolucije, jeete revolucija jeftine energija takoder, ali energija druge vrste jeftine intelektuelne energije



Kralj u prinudnoj penziji Stiven Džobs

— Obavili smo studije koje dokazuju da je mis brži nego tradicionalni načini prolaza kroz podatke ili razne primjene. Jednog dana ćemo možda biti u stanju da ugradimo kolor ekran uz razumnu cijenu. Što se liče previsoke cijene stari novog proizvoda ga čini skupljim nego što će biti kasnije. Što više budemo proizvodili imaćemo nižu cijenu.

• To i jeste ono za što vas optužuju kritičari: pecate zanesenjake — vrhunskim cijenama a onda se okrećete i snižavanjem cijena hvalate ostatak tržišta.

— To jednostavno nije istina. Čim to možemo mi snižavamo cijene. Istina je da su naši računari jeftiniji danas nego prije nekoliko godina ili čak lani. Ali to je također istina i za IBM PC-a. Naš cilj je da dovedemo računare do dosežne miliona ljudi i što ih jeftinije proizvodimo lakše nam je da taj cilj ostvarimo. Ja bih volio kad bi makintoš mogao da košta 1000 dolara.

• Šta je sa ljudima koji su kupili tzv. epl III — dva računara koja ste izbacili na tržište prije makintosa? Ostavili ste ih sa nekompatibilnim i zastarjelim proizvodima.

— Ako žalite lako da govorite onda dodajte na tu listu i ljude koji su kupili IBM PC-a ili PC juniora. Što se tiče računara Lisa, pošto je dio iste tehnologije koristan i za makintosa, on može da koristi softver makintosa i mi na njega gledamo kao na makintošovog starijeg brata iako je u početku bila neuspješna prodaja Lise je probila plafon. Također još uvek prodajemo preko 2000 epla III mjesečno — više od pola starim kupcima. Konačni zaključak je da nova tehnologija ne mora značiti zamjenu stare nego joj se pridružuje. Po delin cipi. Ponekad ona je zamjenjuje. Ali to je onda slučaj kao sa ljudima koji su imali crno bijeli televizor kada se pojavio kolor. Oni su tada zaključili da li im se investicija u novu tehnologiju isplati ili ne.

• Po brzim kojom se stvari mijenjaju zar nece i sam mek biti zastarjelo za nekoliko godina?

— Prije makintoša poslovala su dva standarda — epl II i IBM PC. Ta dva standarda su kao dvije rijeke koje su se urezale u slijepu kanjona. Trebala su godine da se one usijeku u taj kanjon — sedam godina za epl II a čelim za IBM. Ono što smo uradili sa makintošem je da smo u manje od godinu dana silom revolucionarnog aspekta samog proizvoda i svom raspoloživom energijom marketinga koju smo imali kao kompanija biti u stanju da probijemo treći kanal kroz tu stijenu i stvorimo treću rijeku. Treći standard. Po mom mišljenju danas poslovje samo dvije kompanije koje su u stanju da to urade. Apple i IBM. Mozda je to zaista loše ali da se to sad ponovi predstavlja monumentalan napor i ja ne mislim da će to Apple ili IBM uraditi u narednih tri ili četiri godine. Pred kraj osamdesetih cemo možda vidjati neke nova stvari.

• A u međuvremenu?

— Razvoj će biti u tome da se proizvod naprave sve više i više prenosivim da se poveže u mreža da sa ostivare laserski pisači da se dobiju razdvajanje baze podataka da se dobiju veće mogućnosti komuniciranja mozda da se spoje telefon i lični računar.

• Još je dug put pred njim. Neki ljudi su rekli da će makintoš stvoriti ili slomiti Apple. Nakon Lize i epla III akcija Apple-a su strmoglavo pale i industrija je spekulirala da Apple neće preživjeti.

— Da osjetili smo lažinu svijeta na nasim plecima. Znali smo da moramo izvuci zeca iz šešira sa makintošem — ili da nikad nečemo ostvariti snova vezane za sam taj proizvod i za samu kompaniju.

• Koliko je to bilo ozbiljno? Da li je Apple bio blizu bankrotstva?

— Ne na ne. U stvari 1983. kada su sva ova predviđanja nastala bila je fenomenalno uspješna godina za Apple. U 1983. smo gotovo udvostručili našu veličinu. Od 583 000 000 dolara u 1982. prodaju smo u 1983. povacali na oko 980 000 000 dolara. To je skoro sve bilo povezano sa epiom II. Samo se nisu ostvarila nasa očekivanja. Da makintoš nije bio uspješan vjerovatno bi ostali na prodaji od oko milijardu dolara godišnje prodajući epl II i njegove verzije.

• Šta je onda bilo pozadina priča da je Apple gotov?

— IBM je ušao veoma silovito i leg je prešao na njegovu stranu. Proizvođači softvera su se okrenuli IBM-u. Poslelo je jasno svima nama koji smo radili na makintošu da će to rasturiti industriju da će to radikalnisati industriju. A upravo je to i želio da uradi. Da makintoš nije bio uspješan ja bih morao baciti peškir (u znak predaje) jer bi to značilo da moja vizija industrije nije spravna.

• Epl III je trebao biti vas poboljšani epl II ali je bio promašaj od kad je lansiran prije četiri godina. Povukli ste prvih 14000 komada pa i revidirani epl III nije nikad uzlelio. Koliki je bio gubitak na epiu III?

— Beskonačan nerazračunljiv iznos. Mislim da bi IBM da je ovaj računar bio uspješniji imao daleko leži posao da ude na tržište. Ali to je život. Mislim da smo isplivali iz ovog iskustva mnogo jači.

• Još kad se pojavila liza koja je također bila reklamiran neuspjeh na tržištu! Šta se to desilo?

— Kao prvo bila je previše skupa — oko deset hiljada dolara. Uhvatili smo se u kompaniju sa liste 500 najvecih časopisa FORTUNE pokušavajući da prodajemo njima dok su naši korijani da prodajemo narodu. Ono je i drugih problema kašnjenje u isporukama softvar nije ispao tako dobar kako smo se nadali i izgubili smo zalet. A IBM je ušao veoma snazno. Mi smo odlučili da unajmimo ljude za koje smo mislili da su eksperti u marketingu i menadžmentu. Nije to bila losa deja ali na nesreću to je bio tako nov posao da se stvari koje su takozvani profesionalci znali išle samo na uštrb njihovom uspjehu u ovom novom načinu gledanja na posao.

• Da li je to bio odraz nesigurnosti kod vas samih — Stvari su postale krupne i sada se odigrava ključna lopta. Bilo bi bolje da uvedemo u igru prave profesionalce?

— Prisjetite se mi smo imali po 23, 24 i 25 godina. Ništa od toga nismo ranije radili tako da se činilo da bi to bila dobra stvar.

• Da li je vecina tih odluka dobrih i loših bila vaša?

— Trudili smo se da nikad jedna osoba ne donosi sve odluke. U to doba su i tri osoba vodile kompaniju. Majk Skot (Mike Scott). Majk Marakula (Mike Marakkula) i ja. Sada smo to Džon Skuli (John Scullay) predsjednik Apple-a i ja. U ranim danima eko je bilo neslaganja ja bih obično podcinjavao svoj sud onome koji su imali ljudi sa vecim iskustvom od mog. U mnogim slučajevima oni su bili u pravu. U nekim važnim slučajevima da smo išli mojim putem uradili bi bolje.

• Zašto na polju računara dominiraju tako mladi ljudi? Srednji vijek zaposlenih u Apple-u je 29 godina.

— Kako poslaju stariji ljudi se sve laze mijenjaju. Naši mozgovi su naka vrsta elektronskih računara. Vaše misli konstruišu oblika u mozgu nalik na skele. U stvari stvaraia irsjne hemijske odlike. U vecini slučajeva. ljudi bivaju uhvaćani u te oblike nalik na zapis na gramolonskoj ploči i nikad ne uspijevaju da iz njih pobjegnu. Rijetka su osobe koje stvaraju tai zapis različit od uobičajenog pogleda na stvari uobičajenog načina ispitivanja stvari. Rijatkosi je da sretnete nekog umjetnika u njegovim 30-setim ili 40-desetim sposobnog da doprinese nešto izvanredno. Naravno posloje ljudi koji su po prirodi znatiželjni zauvijek mala djeca cijelog svog života ali oni su rijetki.

• Mnoga osobe u 40-desetim godinama će biti zaista oduševljeni vama. No predimo na drugu stvar o kojoj ljudi govore kada sa spomene Apple. Vi imate slican misionarski poriv i u pogledu načina na koji sa radi u Apple-u zar na?

— Ja zaista osjećam da posloji i drugi ulicaj koji mi imamo na društvo sem onog koji čine naši računari. Mislim da Apple ima šansu da bude model Fortune 500 kompanije u kasnim osamdesetim i ranim devadesetim godinama. Prije desal do polnasi godina da sle pitali ljuda da vam saslave listu od pal najuzbudljivijih kompanija u Americi. Xerox i Polaroid bi bili ne avakoj listi. Gdje su oni sad? Danas ne bi bili ni na jednoj listi! Šta se to dogodilo? Kompanije **koko rastu de bi postale entiteti od više milijardi dolara, ne neki način gube svoje vizije.** One umecu mnogo slojeva srednjeg upravljačkog kadra izmedu ljudi koji vode kompaniju i ljudi koji obavljaju posao. **One više nemaju usađen cajaćaj pasije za proizvod koji rade.** Kreativni ljudi a to su oni koji sa žarom brinu o stvarima traba da ubijede pel slojeva manadžera da urade ono za što oni znaju da se traba uraditi. Ono što se dešave u većini kompanija je da se pojedinačno postignuce ne podržava nego baš obeshrabruje. Kreativni kadrovi odlaze i vi završavate sa mediokritetima. Ja to znam jer je tako nsatao

→

15) programer koji je hteo da izmeni svet

Apple je četvrto izbjeglica Apple je sastavljen od izbjeglica iz drugih kompanija To su kranje nadareni individualisti koji doprinose uspjehu a koji su stvarali probleme u drugim kompanijama U svakom slučaju jedan od najvećih izazova po kome traba očarljivali Džona Skulija i mene za pal do desal god na ja da napravimo od Apple-a kompaniju od nevjerovatno velikih desal ili dvadesel milijardi dolara Da li ca ona i lada imali duh koji ima danas? Mi iakočavamo novu tavdoriju Nema modela na koje se možamo ugledati za nake nova koncepte upravljanja koja imamo da bi omogucili lako valik rasl Sloga moramo naci svoj vlastiti put

• Ako je Apple stvarno takva kompanija zašto onda projicirah dvadeseiostruki rast? Zašto ne ostanete relativno mali?

— Način na koji to može funkcionirati u našem poslu da bismo nastavili da budemo jedan od glavnih nosilaca napretka jesta da postanemo kompanija od desal milijardi dolara Taj rast nam je potreban da bismo izdržali konkurenciju Nas vše brina kako da to postignemo nego sama zarada koja je za nas beznačajna U Apple-u ljudi rade po 15 časova dnevno Mi privlačimo drugačiji tip osoba — osobe koje ne žele da čekaju pet do dasel godina da bi se nasao neko ko će preuzeti giganlski rizik za njih Neko ko stvarno želi da se propna malo iznad svoga slaia i zagrize malo u univerzum **Mi smo svjesni da radimo nešto značajna** Mi smo tu ne njegovom početku i u stanju smo da mu damo oblik dalag razvoja Svako je ovda uvjeran da je upravo sada momani kada ulićemo na buduć nosi **Već dio vramana mi samo uzimamo atvari**

„Appia je sastavljen od izbjeglica — kranje nadaranih individualista koji su stvarali probleme u drugim kompanijama'

N v n ja n smo napravili odjela koja nosimo mi ne pravimo nranu niti uzgajamo bilike koja jedemo korisimo jazik koji su razvili drugi koralimo matamatiku drugih društava **Veoma rijetko imamo šanau da stavimo nešto u taj bazen iz koga uzimamo** Mislim da sad imamo lu pritku Ali ne m na znamo gdje ca lo dovesti Mi samo znamo da je lo nešlo mnogo veca od bilo koga od nas ovde

• Da bi ova industrija stvarno procvjetala i da bi to stvarno konstilo potrošačima rekao je jedan ekspart traba da praviada jedan standard

To jednostavno nje istina Insialirati na tome da nam traba jedan standard ja isto kao i kazali da je u automobilskoj industriji 1920 godina trebao jedan standard Da au oni u lo lada vjarovali ne bi bilo nikakvih mnovacija kao što su automatska transmisija neovisno vješenja i si Posljednje što zalimo ja da zamrznamo tehnologiju **Medu računarima „makintaš" je revolucioneran Uopšte za ne postavlja pitanja eko tega da ja tehnologija „mekintoša" superiorna nad IBM ovom Postoji jaana potreba za alternativom IBM-u**

• Zar vaša odluka da ne budete kompatibilni sa tBM om ni u jednom momantu nija bazirala na činjenici da niste željeh da se pokorite IBM u? Jedan kritičar kaže da je razlog zašto mek nije IBM kompatibilan čista arogancija — i da je Stiven Džoba rekao IBM-u Da sa

— Nismo dokazivali svoju muškosi žaljom da budemo različiti

• Pa zašto ste onda drugačiji?

— Glavni razlog je jednostavno laj da je tehnologija koju smo razvili supariornija Ona

„Japanci uzmu nešto što je vec izmišljeno i studiraju ga dok ga u potpunosti na shvate. Iz tog rezumjevanja oni ga ponovo izmišljaju u mnogo rafiniranijoj verziji druga generacije"

mozda ne bi bila lako dobra da smo trabali biti kompatibilni sa IBM-om Naravno lačno je da mi ne želimo da IBM dom nira ovom industrijom **Mnega ljudi misle da smo ludi šte nismo IBM kompetibilni i šte ne živimo pod kišobranom IBM-a** Posloja dva glavna razloga šlo lo nismo uradili prvi je šlo smo mislli i ja mislim da će budicnosi pokazati da smo bili u pravu — da IBM može zalvoriti laj kišobran nad kompanija ma koje proizvode kompatibilna računara i lako ih uništiti Drugi i mnogo važniji — mi nismo išli pulam kompatibilnosli sa IBM-om zbog vizija pro izvoda koja je pokralačka snaga ova kompanije Mislimo da su računari najizvanrednija alatka sa kojima se čovječanstvo arelo i mislimo da su ljudi u osnovi korisnici alata Tako ako budemo u stanju da mnogo računara, damo velikom broju ljudi lo ca dovesli do kvalitelivne razlike u svijetu Ono šlo mi želimo je da od računara stvorimo dio kućnog namještaja i da ga dovadamo do dasalina miliona ljudi Jednostavno lo želimo da uradimo A lo ne bismo mogli uradili sa lakućom generacijom IBM tehnologija Sloga smo morali da izradimo našlo drugo Zalo smo proizvali mekintoša

• Kao i kompjuterska industrija automobilska industrija je bila američke industrija koju smo skoro izgubili — preuzeli su je Japanci Ima mnogo priče oko toga da amerićke kompanije proizvodeći poluproizvodnika gube tlo pod nogama zbog Japanaca Kako cete vi izdržati?

— Japan ja veoma inleresanlan **Neki ljudi miala da oni kopiraju atvari** Ja lo više ne mislim **Ja mislim da oni ponovo izmišljaju atvari Oni uzmu našto što je već izmišljena I studiraju te dok ga u potpunosti ne ahvata** U nakim slučajevima ga ahvale bolja nego originalni pronalazač **Ia tog rezumijevanja ani ga ponova izmišljaju u mnogo rafiniranijoj verziji druge generacije** Ta strategija funkcioniše samo onda kada se ono na čemu oni rada na mjenja brzo — industrija stereo uredaja automobilska industrija su dva primjera za lo Kada se cil brzo kraće lo im ja jako leško jar ciklus rainvancija traje nakoliko godina Dok god sa dalinicija ličnog računara naslavlja mijanjati brzinom kojom se Iranulno mijanja oni će imati laška vramana Jadnom kada se brzina promjena uspori Japanci će bacili sve svoja snage da se probiju na ovo tržište Oni apsolulno žala da dominiraju ovim poslom o loma nema aumnja Oni na lo gledaju kao na nacionalni priorilal Mi mislimo da ce za čeliri do pal godina Japanci napokon otkriti kako da naprava česlil računar Ako želimo da zadrzimo primal u ovoj industriji imamo čeliri godina da postanemo avjetski proizvodači

• Kako planirate da to ostvarite?

— U vrijama kad smo projektovali makinlo ša mi smo takoder projektovali i mašinu koja pravi mašina Potrosili smo 20 000 000 dolara da izgradimo najautomatizivaniju tvornicu industrija računara Ali lo nije dovoljno Umjesto da amortizujamo našu tvornicu za aedam godina šlo bi uradila većina kompanija mi ćamo ja amortizirali za dvije godine Bacil ćamo ja do konca 1985 i sagradili cemo drugu a i nju cemo olpisati za dvije godina i bacili je i lako ćemo za tri godina od danas radili na tracoj automatizivanoj tvornici To je jedini način da učimo dovoljno brzo

• Razgovarajmo o softveru Koja su revolucionarne promjena u razvoju softvera u posljed njih nekoliko godina?

— Sigurno da unošenja programskog jazika u mikroprocesorskr čip predstavlja napredak VisiCalc je bio napredak jar je to bilo prvo atvarno korišlenja računara za poalovna avrha gdja su poslovni ljudi vidjali opipljive korisli

Prija loga ste trabali programirati svoja vlesilte aplikacije a broj ljudi koji je voljan da programira je mali — svega jedan poalo Onda je došao Lotus — kombinacija dobrog spreadshael-a (matrična tabela) i gralickog programa Procesor laksia i baza podalaka Lotus-a sigurno nau ono najbolje šlo čovjek može kupiti Stvarna vrijednost Lotus-a ja kombinacija labale i gralike u jednom programu i mogucnosi brzog skakanja sa jednog na drugo Slijedeći prodor aa dešava sada zahvaljujući makintošu koji je donio tehnologiju liza uz naravnu cijenu Tu ima a bice jos više revolucionarnog softvera Pravi prodori se mogu cijenlir nakoliko godina nakon šlo su se dogodili

• U kom pravcu vi vidite razvoj računara i softvera u bliskoj budućnosh?

— Za sada mi prilično korislimo računare kao dobre sluga Tražimo od njih da nam našlo uradе — tražimo da urada neku operaciju kao šlo je obračun labela (spreadshsel a) tražimo da prihvala naše kucanja po labalaluri i od loga naprave piamo i oni lo rade prilično dobro Dakle vidjecela ave više i više parfakcija u domanu — računar kao sluga Ali slijedeća atvar će bili računar kao vodič ili agani A lo znači da će sve više bili u stanju da anticipira ono šlo želimo Ećemo u stanju da od računara tražimo da prati događaje umjesio nas i kad se odredani uslov ispuni da povuća okidač — da obavi neke akcije i da nas obavijesti o loma

• Hocemo li biti u stanju da sve to obavimo na mašinama koje danas imamo? Ili cete nas opteretiti novim računarima?

— Sve? To bi bila opasna izjava To na znam Makintoš je u svakom slučaju dizajniran lako da su se li koncapi imali u vidu

• Govoriti smo o toma šta vi vidite u bliskoj budućnosti a šta je sa dalekom budućnosti? Ako počnata da zamišljate neke od načina na koje će računari izmjeniti naše živote šta onda vidite?

„IBM je ušao veoma silovito i teg je prešao na njihovu stranu Svima nama koji smo radili na 'mekintošu postalo je jasno da ce to rasturiti industriju'

— Kada sam se vralio iz Indija pilao sam sa šta je najvažnija atvar koju sam naučio? Mislim da je lo da zapadno racionalno razmišljanja nija urodena ljudska osobina nago da je lo naučena sposobnosi Prije mi nije padalo na pamal da mi ne bi ni razmišljali na odredani način da nas neko nija naučio da lako razmišljamo A ipak lo je lako Očigledno jedan od velikih izazova obrazovanja je da naa nauči kako da mislimo Ono šlo olkrivamo ja lo da će računari ulicati na kvalilal razmišljanja lim viša šlo više naše djece buda imalo le aprave na raspolaganju Ljudi su korisnici alata Ono šlo je žalila navjarovalno za knjige je lo da možala čilali šla je Aristolel napisao Na lrabe vam profasorova nlarpretacija Aristolala Naravno možala i lo dobili ali možala pročilali lačno ono šlo je Aralolal pisao Taj direktni prenos misli i idaja a ključni kaman koji zgradu društva čini onakvom kakva ona jesla Ali problem sa knjigom je laj da ne možete da Aristolelu postavila pitanja Mislim da ja potencijalna mogućnosi računara da na naki način uhvali osnovni bazni princip nakog iskustva

Priredio: Domagaj Bačić

Periferijska oprema

svetlosna pera

Pokazivanja prstom predstavlja najjednostavniji način da nekoma ko nas ne razume pokažemo ono što želimo. Za dobar broj korisnika računar će do kraja života ostati „crna kutija" s kojom će se sporazumavati isključivo preko tudih programa. Pošto računari još uvek nisu dovoljno usavršeni da bi zapazili i ahvatili dodir nečijeg prsta, izmišljena je svetlosna olovka — neka vrste „elektronskog prsta" — koja komunikaciju sa računarom pojednostavljuje do krajnjih mogućih granica. Svetlosna pera su, osim toga, i nezamenjiva alatka za ljubitelje crtanja.

Prema rezultatima poslednje ankete američkog časopisa COMPUTE samo 6 odsto ljubitelja računara poseduje svetlosno pero. To je nesrećna okolnost jer ono pojednostavljuje korišćenje programa i najviše odgovara upravo korisnicima računara koje programiranje ne interesuje već koriste opcije iz gotovih programa. Svetlosno pero osim toga, vezuje pažnju kod dece. Ako ste roditelj sa minimalnim ili osrednjim programerskim umećem možete nabaviti gotove programe za obučavanje dece koji se koriste uz svetlosno pero. Za korišćenje ovakvih edukativnih programa deca ne moraju da znaju čak ni da čitaju, jer izbor iz menija koji je prikazan u slikama vrše jednostavnim dodirivanjem željene slike.

Zašto nam treba . . .

Svetlosno pero je nezamenjivo orude kada treba brzo locirati slučajnu poziciju na ekranu. To može biti unošenje koordinata, stavljanje oznaka, crtanje linija, izbor iz menija i slično. Upotreba kursora ili još gore unošenje koordinata sa tastature sporo je i zamorno a svetlosna pera su brza jednostavna za upotrebu vrlo precizna i posebno pogodna za grafiku. Njima možete crtati ili pokazati na ekranu odgovarajuću opciju iz grafičkog programa. Danas se mogu nabaviti ozbiljni aplikacioni programi — tekst procesori i programi za obradu tabela koji umesto ukucavanja instrukcija omogućavaju njihovo unošenje pokazivanjem svetlosnim perom. Jedni programi omogućavaju da se svetlosno pero koristi za kreiranje linija, krugova i kvadrata i popunjavanje tih geometrijskih slika raznim bojama a drugi da svetlosno pero koristite za definisanje grafičkih simbola ili karaktera ili konstruisanje komplikovanih geometrijskih slika. Mada je za igre pogodniji džojstik, u zadnje vreme eva je više igara na tržištu koje se igraju uz pomoć svetlosnog pera — to su pre svega igre koje zahtevaju pomeranje objekate po ekranu na primer, šah. Postoji i nov proizvod u domenu igara za svetlosno pero — svetlosna puška. One je već raspoloživa za spektrum i komodor računare.

. . . i kako radi

Svetlosno pero čita poziciju na ekranu monitora ili televizora. Pri dodiru ekrana elektronika u paru i računar zajedno izračunavaju koordinate te tačke. To je moguće zbog načina na koji se kreira displej na katodnoj cevi (tipu koji je u televizorima i monitorima). Tanak snop svetlosti kreće se ekranom s leva na desno i odozgo na dole. Kod evropskih televizora on prede preko 625 linija displeja 50 puta u sekundi. To jurenje svetlosti je prebrzo da bi ljudsko oko registrovalo pomeranje pa imamo iluziju da gledamo stabilnu sliku. Svetlosna pera sadrže svetlosni senzor pri vrhu. Taj senzor obično svetlosno osetljiv tranzistor ili diode registruje svetlosni trag u momentu proleska. proizvodi elektronski signal pojačava ga i šalje u računar. Pošto računar kontroliše displej na ekranu on „zna gde se u tom momentu svetlost nalazi pa može da izračuna x i y koordinate tačne pozicije vrha svetlosnog pera.

Modeli svetlosnih pera se razlikuju po mnogim karakteristikama — po tipu svetlosnog senzora položaju senzora, mestu na kome se nalazi pridružena elektronika da li daju različite režime reda da li i kako pokazuju kada su registrovali signal i najzad za koji tip računara su primenjivi. Za korisnika je uz cenu, najznačajnije da zna osetljivost i rezoluciju svetlosnog pera.



Crtanje bez muke. Optičko pero znatno olakšava život ljubiteljima crtanja na računaru

Osetljivost . . .

Pod osetljivošću svetlosnog pera podrazumevamo intenzitet svetlosti ekrana koju pero može da registruje. Najbolje su ono koje mogu da registruju veličinu ako ne i sve odje sa dobro podešenog kolor televizora dok svetlosna pera slabijeg kvaliteta registruju samo jaču svetlost. U dosta slučajeva je i to dovoljno ali dovodi do ozbiljnih ograničenja u grafičkim aplikacijama kod kojih pozadina može biti i tamna.

Jedan od problema se svetlosnim perima je nelepravne detekcije. Do nje dolazi kada je osetljivost uvećana velikim pojačanjem signala se svetlosnog senzora. Prateća neosetljivost može dovesti do toga da i venekranski izodri svetlosti pe čak i dnevne svetlost deluju na pero. Skupja pera koriste razna metoda osiguranja od tih izvora svetlosti racimo kombinacije svetlosnih senzora i svetlosnih filtera. Mogu se upotrebiti i elektronski filteri. Svatlost se

➡

ekrana dolazi 50 puta u sekundi Jednostavno elektronsko kolo propušta ka računaru samo svetlosti sa tom frekvencijom

. . . i finoća crtanja

Rezolucija se odnosi na zonu ekrana koja se detektuje Ona može da varira od jedne tačke (piksela) do grupe karaktera kod najlošijih pera Rezolucija svetlosnog pera zavisi od tipa senzora, brzine reagovanja i metode kolimacije (sakupljanje svetlosti) Visoka rezolucija je važnija za crtanje nego za registrovanje Pero se visokom rezolucijom može registrovati manju zonu pa je stoga preciznije

Razna pera imaju različitu dokumentaciju i softver koji se uz njih dobije Da b svetlosno pero uopšte radilo potrebna mu je softverska podrška Neki proizvodeci obezbeduju kasete se demonstracionim programima dok drugi deju samo listinge da bi ilustrovali način korišćenje svetlosnog pera

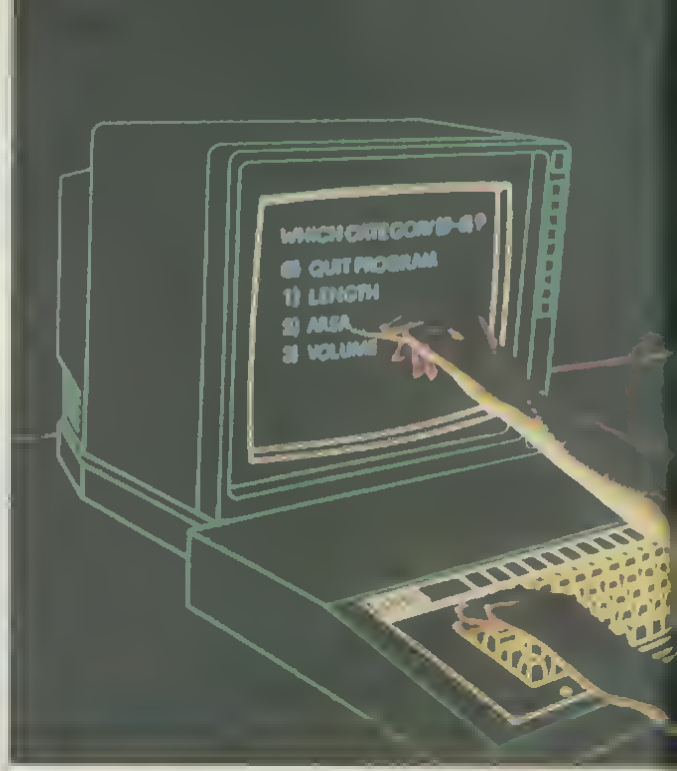
U „mekintoš" stilu .

Kao i kod drugih vrsta specijelnog hardvera i svetlosna pera se odgovarajucom programskom podrškom prvo su bila raspoloživa za skuplje i za nes nepristupačne računere kao što su epl II IBM PC i PCjr eli danas ima sve više tipova za komodor 64 i druge kod nes popularne računare Predstavicemo vam softverske pakete koji se mogu koristiti uz Edumete Light Pen i Gibson Light Pen

Edumete Light Pen, koji nudi Futurehouse ima sve neophodne veze i spreman je de se utekne u džojstik port komodore Tu su i brošura sa instrukcijama i softverski paket se raznim programime ze svetlosno pero Softver sedrži program za crtanje se visokom rezolucijom disk utility muzički program i igru 3D Tic Tac Toe Rutine ze

Da vam život bude lakši

- Svetlosna pera moraju bit savršeno čista Svaka nečistoca ulice na svetlosti koju pero treba da sakupija Postoji dosta čelkica za čišcenja pera ili ekrana baziranih na alkoholu
- Pero zamračuje bar deo ekrana kada ga koristite To je hendikep jedino ako program ima veoma mnogo dalalja na ekranu
- Pre kupovine se uverite da za vaše potrebe ima dovoljan džojstik Na zaboravite da za crtanja po vertikalnoj površini ekrana treba mnogo više truda nego po horizontalnoj površini hartija Ako vam cena nije problem koristite grafičku tablu
- Ako se vaše pero ponaša nepravilno ili je relativno možda skupija previše svetlosti iz sobe. Pokušajte da spustite zavesa ili ugasite neku sijalicu
- Možda treba da podasite osetljivost ili kontrast na talavizoru da bi paro radilo kako treba
- Pri biranju opcija iz menija ne šetajte pero okolo već ga usmarite pravo na željenu opciju u suprotnom možete dobiti pogrešnu tačku i pogrešan meni
- Neka novija pera otkrivaju koja je boja na ekranu Ako imate takvo pero proverite da li vaš softver kaže peru koju boju da registruje. Možda ćete morati i da podešavate boju na ekranu sve dok na bude potpuno odgovarala peru



Pero umaste miša optičkim slavkom može se najbrža dovaci proizvoljno odabrana pozicija na ekranu

crtanje omogućava de se crta u grefičkom režimu visoke rezolucije Nudi se 16 boje se opcijeme brisenje ekrana ili uklenjanja linije Disk Utility je opštenamenski DOS (Disk Operating System) program koji nudi opcije displeje direktorijume i mogućnost birenje punjenja i stertovenje progreme pomocu svetlosnog pera Uz to može se inicijelizovati, velidizovati ili formetireti diskete prikazeti direktorijum i izbrisati bilo koja datoteka ili program

. . . i nešto malo siromašnije

Za spektrum ćemo predataviti tri vrste svetlosnih pera, čije se cene kreću izmedu 15 i 20 funti

Svetlosno pero koje je ne tržištu izbecio Custom Cebels Internetionel najjeftinije je i se najmenjim mogućnostime eli korektno redi sa avekom opcijom Opcije uključuju crtenje svih geometrijskih oblike i što nameju osteli proizvodi crtenje poligona, za koji semi birate broj strane Ovo pero neme meni opciju ni mogućnost slobodnog crtenja eli eko vem ove dve stveri nisu bitne bićete sesvim zedovoljni njegovim kvelitetom

Svetlosno pero TROJAN neme najbolju softversku podršku Kelibracione rutine koje provereve de li se slaže ono što vidite sa onim što sa šalje u računer je neprecizne i čini se bar u demonstraciji menje da se pozicije pere čita prebrzo Rutina slobodnog crtenja radi tako sporo da je praktično neupotrebliva Možete crteti krugove linije i ostele progremirane figure ali je pero suviše nepreciznо da bi to učinilo lekim Kad se sve sabere više se isplati kupiti jeftinije pero čije sve opcije korektno rede

DK tronička MK 4 pero je sesvim druge (najskuplja) priče Broj nije bez znečaje MK 3 verzija je semo neznetno bolje od TROJAN SP ali MK 4 dozvoljeve razumno crtenje slobodnom rukom Jedino je problem mirno držati ruku na gotovo vertikelnoj površini ekrena Osim kreirenje linije pravougeonika i krugove možete sečuveti do pet ekrena i staviti ih u rotecije od veome brzih (pseudoanimecije) do izuzetno sporih Rutine slobodnog crtanja zahteva melo napora ali on se isplati

Pen Music omogućeve svirenje dodirom svetlosnog pere po raznim tečkema ekrene Prikazuje se zone od 12 note — kompletne muzička skale u bilo kojoj od 8 okteve

Mede su svi programi upotrebljivi moremo napomenuti de imaju i nekoliko ograničenje Korišćenjem Disk Utility mogu se puniti samo bejzik programi Opcije formetizovanje diskete na omogućava izbor imena diskata Muzički program na omogućave biranje filtara i jačine zvuka Crteža krairane ovom rutinom za crtanje ne možete de sačuveta na traci ili disku i de ne nabrojemo delje — i najboljim peketime mogu sa neći mene





Optika u akciji Čak i kod najskupljih modela optičkih olovaka crteži su još uvek daleko od savršenstva

Jeden od vodećih dizajnera mikroračunarskog grafičkog softvera Stiven Gibson, pripremio je paket Gibson Light Pen za komodor računera U ovom paketu nalaze se programi Pen Painter Pen Animator i Pen Musician Prvi od njih je kolor grafički program koji omogućava slobodno crtanje, crtanja prevlnih geometrijskih likove i deje obresce za punjenje bojom Ovej softver omogućeve de se izbor vrši iz menije u -mekintoš stilu — birenjem slićica Drugi program predstevlje uvod u animirenu kompjutersku grafiku i omogućava de se kreire do 20 delova enimecije koji mogu de jure po ekranu dovoljno brzo de bi se stvorio utisek kretenja Treći program uvodi u muzičku kompoziciju i dozvoljeve de postevite editujete i izvodite note na muzičkoj skeli pomoću svetlosnog pere

Izbor — uvek vaš

Ime više fektora koje trebe imeti u vidu pn izboru svetlosnog pere Mede neke evetlosne pere teoreteki imeju visoku rezoluciju u praksi nisu dovoljno precizne Neka su čak toliko grube de im se linije razbijeju u mnoštvo piksele po ekrenu Princip je i ovde — dobiješ koliko platiš Cene varireju od cene za tipično memorijsko proširenje do cene jettinijih računere

Neravno kao i kod ostelih prizvode (ne kod nes) cene stelno padeju

Neke, obično skupije pere imaju još dve korisne osobine — proizvode signal obeveštenje da je pero registrovalo svetlost i signel de je svetlost predeta u računar Svetlosna pera obično koriste „svetloono emitujuću diodu ze obaveštenje da je registrovene ispravne svetloet To nije od znečeja ako nem pero treba samo ze crtanje eli pomaže za čitenja pozicije se ekrena ili izlaz iz menije

Svetlosne pera često imeju prekideč koji obezbeduja de se informacije šalju u računer eemo kade želimo što je korieno jer smenjuje mogućnost slučejnog ili pogrešnog čitenje (pogrešan izlaz iz menije npr) Sem toga, imamo više kontrola ned čitenjima koje vrši pero Postoja razni tipovi prekideča Neke od boljih pera imeju minijeturne prekidače pri vrhu koja reeguju tek eko pero dodirne ekran Međutim bez obzira šte se piše o mogućnostime pera preporučujemo de pri kupovini obavezno isprobate svetloono pero i proverite de li radi oneko kako želite

Najvažnije je de proverite de li svetlosno pero radi s vešim računerom Spektrum i ZX 81 sigurno zahtevaju interfejs koji se po previlu ugraduje i u seme pere Ne zabore vite ni softver Ako kupite svetlosno pero koje nije dizejnireno za veš računar nije leključeno de ćete morati sami de ge pišete ili kupujete novi eoftver de bi uopšte proradilo

S obzirom ne veliku primenljivost za igre i ozbiljne eplikacije brzinu i jednostevnost upotrebe i nisku cenu populernoet svetlooenih pere reste iz dene u den Cene će i delja padeti i biće više respoloživog softvera Ze korisnike koji žele de uneprede igrenje kompjuterskih igera eli i ona koji hoće de udobnije koriste ozbiljne programme svetlosno pero je ideelen dodetek računeru

Vladimir Iiljevski
Nevanka Spalević

haker *Umetnost programiranja*
na usijanom limenom krovu



Osnovni problem u testiranju je pogrešna definicija koja se obično izlazava na jedan od sledećih načina Testiranje je proces demonstriranja da nema grešaka u programu Smisao testiranja je da se pokaže da program radi korektno ili Testiranje je proces sticanja uverenja da program radi ono što treba da radi Sve ove definicije imaju jednu jedinu manu totalno su pogrešne! Pozabavimo se analognom situacijom u kojoj bolestan pacijent dolazi na laboratorijsku preiragu Ako analize ne otkriju poreklo problema sigurno nećemo reći da je postupak bio uspešan vreme je izgubljeno pacijent je i dalje bolestan a uz to polako počinje da sumnja u kompetentnost dijagnostičara Obratno ako pretrage pronađu ćir na žalucu može se pristupiti lečenju (Mogućno je dakako da pacijent boluje i od drugih bolesti)

Ova analogija nas navodi na ispravnu definiciju testiranja koja glasi Testiranje je proces izvršavanja programa sa ciljem da se pronađe greška! pri čemu se podrazumeva da je program bolestan tj da sadrži greške Sa izuzetkom retkih hardverskih grešaka sve ostale greške u programu prave ljudi — programeri pa uspešno testiranje zavisi od prilaza problemu Ako je cilj pokazati da u programu nema grešaka onda ćete verovatno uspeti da pokažete da ih ima malo ako je suprotno tome cilj pokazati da su greške prisutne onda ćete verovatno naći njihov dobar deo! Na žalost kod tole vecih programa nikada nećemo biti sigurni da smo pronašli sve postojeće greške

Crne i bele kutije

Pristup crne kutije Baš ma briga kako je program napisan propustiću sve moguće ulazne podatke i ako program dobro radi — sve je u redu! Pristup bela kutija je totalno suprotan Napraviću taman toliko testova da se svaka grana programa izvrši barem jednom Najvažnije je proveriti logiku programa! Nijedan od ova dva pristupa nije mogucno strikino provesti u praksi Ulaznih podataka ima u svakom ma kako trivijalnom programu praktično beskonačno Kako recimo testirati jednu akcionu igru u kojoj se sva dešava u realnom vremenu? U njoj bi trebalo testirati na samo sva mogućna kombinacija ulaznih podataka nego i sve mogućna situacija u toku igra — a broj takvih situacija je astronomski

Jasno je da sva ulazna podatke nikad na možemo proveriti u praksi — ekonomičnost je osnovni problem u procesu testiranja programa Svaki test treba de predstavlje čitevu klasu ulaznih podataka teko de sa minimumom podataka dobijamo maksimum informacija o ponašanju programa Ni pristup bela kutija nije ništa bolji u svom ekstremnom vidu Broj mogućih putava u iole složenijem programu rasta es faktorijalom to jest strašno brzo Štaviše i kada bi smo bili u stanju da izvedamo potrebna bilione tsetova u nako dogledno vreme čak ni onda ne bismo mogli tvrditi da u programu nema greške Šta ako je programar napiseo prpgram za računanje kvadratnog korena — a tražio se program za kubni koren? A tu je i problem osetljivosti podataka naka grana programa izvršavaju sa za datu tačnost ulaznih podataka ali ako podaci izađu iz predviđanih granica — onda se izvršavaju sasvim drugi delovi programa Najkraće rečano testiranja je praktično nerešiv problem ma kako se trudili u procesu pisenja programa, greška će biti prisutna a kada su već jednom tu — možamo ih testiranjem uklanjati ali nikad nećemo biti sigurni da su zaista sve uništena Zato je važno da se pridržavamo sledećih fundamentalnih principa testiranja

Principi . . .

• Dobar je onaj test za koji je vrlo verovatno da će otkriti grešku u programu dok je test koji demonstrira pravilan rad programa neuspešan Cela aktivnost testiranja je usmerana na poboljšanje kvalitata programa, tj na poboljšanja pouzdanosti programa u toku eksploatacija Povećanje pouzdanosti postiže se nalaženjem i uklanjanjem grešaka iz programa Stoga i na traba previti test-primera koji pokažuju de program dobro radi- već suprotno de program radi pogrešno!

• Programer koji je piseo program na bi trebalo i da ge tsetira! Projaktovanje i pisanje programe je vrlo kreativne ljudska delatnost e tsetiranje je rušilački proces u kome sve napisano treba podvrći sumnji i kritici (Da li ste ikeda čuli za pesnika koji je napisao kritički osvrt na svoju najnoviju pesmu?) Ako je programiranje umetnost pisanja asaja u izvršnom obliku onda je testiranja pisanje kritičkog prikaza — a kritika sa objavljuju dva-tri dana posle premijere kada se duhovi smire Ako želite zaiste kvalitetan program zamolite svog prijatelja da damolira veša potencijelno ramek-delo Ako ipak hoćete (ili morate) sami da testirate svoj program nemojte to radili istog dana kada ste napisali i ukucali program u računar Namarno se odvojite od mašine na dva-tri dena ohladite se naljutila se na svoj program i tek onda počnite sa testiranjem

• Svaki test mora sadržati i opis očakivanih izlaznih podataka ili razultata Jedna od najčešćih grešaka u tastiranju upravo potiča od ignorisanja ovog pravila Očakivane razultate moramo izračunati ranija ineće dovodimo sebe u situaciju da vidimo ono što hoćamo da vidimo a ne ono što jesle Izvršavanje programa baz unapred poznatih razultata nije testiranja već — eksperimentisanje

• Kućni i tićni računari, baš kao i rad sa larminalima u sistemu tajm-šaringa su kontraproduktivni tastiranju Isuviše je lako upasti u začarani krug u kojem vi samo sedila i pritiskala dalova tastature izvršavajući program opat i opat a sve u strahu da računar ne stoji Nikad na stižala da napravite valjani test-primar a najveći problem je što se testiranje obavija u letu testovi iščezavaju baz traga pa posle svake nove izmane u programu traba ponovo izmišljati već stvorene no izgubljene testove

• Testove treba dokumentoveti i čuveti u obliku daloteke na spoljnoj memoriji Time štedimo trud na kreiranju testova i istovremeno osigurevemo da program posle svake izmena buda testiren na lali nećin

Faze testiranja zauzima više od 50 odsto vremena u svakom programerskom projektu. Godišnji obrt računaraka industrija meri se dasatinsma milijardi dolara, pa bi se mogio pomisliti da je testiranje dobro izučana aktivnost ali nija tako Problem ja kako u samoj prirodi process testiranja, tako i u nedovoljnoj obučsnosti programers za ovaj deo svog posla Većina testova sročena je tako da po svaku cenu dokaže da program radi, dakle po principu „hajde da se pravimo blesavi" — „maločas ja odlično radio ne znam šta mu sad bi" Cilj svakog pravog testa ja, međutim, da pokaže da program ne radi kako traba

• Testirajte program kako pomoću tačnih tako i pomoću netačnih i neprihvatljivih ulaznih podataka Ovo je najbolji način da zaštite program od naivnih korisnika pa i od samog sebe kroz šest meseci i vi cete biti samo korisnik sopstvenog programa!

• Datajno izučite ave razultata svakog testa I najbolji testovi na vrada ako ne interpratirata njihove razultata Ovo pravilo je toliko očigledno da ga na bi trabalo posabno izdvajati — samo da sa tako često na pranebragava u praksi!

• Nikad na menjajta program da bista sebi olakšali testiranja! Racimo da naki ciklus od 100 prolaza amanjita na 10 — da bi ste brža dobili razultate testa U tom slučaju zaista vršita testiranja — ali na avog nego nakog drugog programa!

• Nija dovoljno uveriti sa da program radi sva što traba da radi traba sa uveriti i da program ne radi nešto što na bi trabalo da radi! Primer takvog nažaljanog bočnog atakta ja procesor reči koji od vas obavezno zahtava da tekst animita na traku pre stampanja

• Graška nailaze u gruparna! Ovaj fanoman sa stalno iznova pojavljuja u praksi mada još niko nija dao zadovoljavajuće objašnjanja zbog čega ja to tako Jednostavno naki dalovi programa podložniji au graškama od ostalih Poznavanja ovog principa daja nam važno praktično uputstvo za testiranje kad naidata na jednu grešku potražita u njanoj okolini bar još jednu ili viša grašaka

. . . i vrste testiranja

• Testiranje modula Modul je nazavisna caline unutar programa sa jasno određenim ulaznim i izlaznim veličinama Testiranje module počinja upravo proučavenjem ulazno/izlaznih podataka a nastavlje se generisanjem testova samo na osnovu specifikacija module Zatim proveravemo tekst modula i ubeđujemo se bez efektivnog izvršavenja programe da odebrani testovi pokrivaju ave interasantne grene modula Na primer, za avaki ciklus mora postojeti bar po jedan test u kojam sa ciklus izvršava maksimalan broj puta Na kraju proverevemo de ti postoje putevi u modulu čija izvršenja zavisi od posebnih vrednosti ulaznih podataka i dodajemo nova testove u tom smislu ako je potrebno

• Funkcionalno testirenje je proces nalažanje razlika između programa i njegove specifikacije tj onog kako je program zamišljan da radi Najbolja je puatiti krajnje korisnike da isprobavaju program neko vreme oni koriste program na krajnja originalne načine, njih ne interesuje kako je program napisan — ako radi bez problema

• Sistemsko testiranje ima za cilj da uporedi program (ili sistem programa) sa početnim ciljem pisanja programa Sistemsko testiranja programa pokušava da prikaže kako sva program ne postiže svoja zamišljene ciljave Jasno sistamsko testiranje ja nemoguće ako projaktant sistema nije zapisao prvobitne ciljava pravljanja programa Najprostiji i najsigurniji postupak za sistamsko testiranja je poredanja osobina programa sa priručnikom za korsnika često je mogućno i bez izvršavanja programa uočiti nasiaganja u ovom domanu

• Količinski tast ispituje ksko program radi sa maksimalnim mogućim brojam podataka tj cilj ove vrste testiranja ja da pokaže da program na može da se izbori sa pravalikim količinama podataka Na primar šta as dešava sa procesorom reči kad ispunimo calu memoriju takstom ili kad u nakoj avanturi posatimo sve lokacije?

• Test načina korišćenja pokušava da otkrita graška u komunikaciji program/čovak Na primar da li su poruke programa prilagodene inteligenciji uzrastu i obrazovanju krajnjag korisnika? De li korisnik mora stalno da pralazi iz malih u valika slova? Da li postoji praviša opcija ili naprotiv da li postoje opcija koje se ne koriste? Da li program javlja graška na razumljiv način? Itd

• Test kontiguracija Da li vaš program radi i sa džojstikom? Da li procesor rači radi sa raznim vrstama štampača raznim pračnicima diskova raznim disk-formatima? itd

• Testiranja oporevijenja sistema De li procesor reči brže sav tekst ako je taster Reset bio pritisnut? De li računar brže calu memoriju posle pritiska na•Raset?

• Testiranje dokumentacije De li program radi sve što piše u priručniku? De li priručnik opisuje kako se koriste sve mogućnosti softvere i/ili hardvara?

• Test instaliranja sisteme Mnogi komercijalni programi se prilegodevaju svojoj okolini što je uslov da sa prodaju na rezličitim računarima Cilj ovog testa je da otkrije ne koje sa sve sisteme dati progrem ne može instalireti

Proverama nikad kraja

Ne ovo pitanje teško je deti precizan odgovor Videli amo da je nemogućno istestireti program ne slžu tj efaktivnim izvršavanjem programe za moguće ulazne podetke ili sve moguće grene unuter programa Ma kako ae trudili senka aumnje uvek ostaje jer neme načina de se dokaže da je greška koju smo tek otkrill baš posledrija u celom programu Programen obično prekidaju testiranje kada program izvrši sve test-primere bez greške Ovakav pristup potiče od pogrešnih definicija testiranja ali na sreću postoje bolji (mada ne i savršeni) kriterijumi

Jedan takav kriterijum može da bude sledeće testiranje modula je završeno kada su neuspešni svi test-primeri koji su nastali pokrivanjam logičkih uslova u modulu kao i analizom graničnih vrednosti ulazno/izlaznih podatake Ovakav kriterijum ja dalako od savršenog jer zavisi od subjaktivna procena i iskustva osobe koja vrši testiranja Najvaci nedostatak je tu ipak odsustvo merljivog cilja Kako je cilj tastiranja nalaženje grešaka logično ja uzeti broj pronadanih grešaka u datom vramanu kao izlazni kriterijum iz procasa testiranja Na primer test modula nije završen sve dok na pronademo 3 graška ili dok ne prode dva dana — koje god od ta dva se dogodi kasnije

Sad ja logično postaviti sledaće pitanja (1) Kako procaniti ukupan broj grešaka u programu? (2) Koji deo postojačih grašaka se realno može otkriti kroz testiranje? (3) Koliko,grašaka je napravljeno u tazi projak tovanja programa a koliko grašaka otpada na kodiranje? Na prvo pitanja odgovara nam uglavnom iskustvo stačano kroz protekla testiranja Uzima se da je industrijski prosak 4—8 graška na 100 linija programa Odgovor na pitanje (2) ja ponešto proizvoljan i zavisi od priroda programa kojeg traba testirati Odgovor na pitanje (3) je jpak najteži Moža se uzati da u velikim programima samo oko 40% grašake nastaju u procesu kodiranja a sva ostala nastaju u ranijoj fazi projektovanja Gotovih formula za kraj testiranja nama pa je najbolje postati umetnik u pravljanju kompromisa treba koristiti prvo kriterijum završetka te ste module a zatim neku razumnu kobinaciju gornja tri kriterijuma

Sve u svemu, testiranja je izuzetno kreativen i intelektuetno izazoven zadatak Da li to znači da samo geniji mogu da testiraju programe na kvalitetan način? — Sigurno ne! — jer ipak postoje metodi za generisanje testove U okviru pristupe crne kutija to su razbijanje po klasama akvivalencije analize graničnih uslova grat uzroka i posledice keo i prosto pogađanje teških test-primera U okviru pristupa bala kutija razlikujemo metode izvršenje svaka naredbe izvršanje sveke odluka pokrivenje svakog uslova pokrivanje svake odluke i uslova i na kreju višestruko pokrivenje uslove U praktičnoj primeni trebalo bi koristiti skoro ave ili baš sve spomenute metode generisanja test-primera jer nijedan od njih sam ze sebe na obezbeđuje potpun uspah Idealno bi bilo prvo napraviti testove po matodama pristupe crna kutije a zatim dodeveti testove koristeći metode bele kutije Ali o tome u sledećem broju!

Duško Savic

21/haker na usijanom limanom krovu

Smešna strana računara

računarenje i računsko razgovaranje

Očito je da su računari ušli u svakodnevni život — ako ne kao predmet ono bar kao pojam. Ali to više nije ona reč koje je označavale misteriozno i opskurno mašinu za koju se samo znalo da čini čuda. Računari su danas čvrsto ugrađeni u trend i kao takvi predstavljaju obaveznu lektiru za sve ona koji pretenduju da budu „u toku" Kako staći računarski manirizam i u svakom razgovoru zvučati kao da si provao pet godina na Massačusetsu specijalizujući Tjuringove nauke? Evo oveko.

Najomiljenija tema računarskih priučenih erudita su naravno tipovi računara. Svaki od modela i tipova ili vrsta ima obaveznu pretpostavke i sulkse zavisno od toga da li je čovek koji govori navijač ili protivnik tog računara. Sinkler ZX 81 je lako smešan igračka i minijatura. To bi još i imalo smisla da te reči masovno ne koriste ljudi koji ni jednom računaru pa ni 80 kecu nisu prišli bliže od dva metra. O spektrumu i komodoru 64 se sve zna. Spektrum je isto igračka ali i pocalnički idealan simpatičan jaftin i dobar za vežbu. On ima nadovoljan ROM i odvratnu tastaturu a predstavlja smrt za mašinac.

Komodore ima dosta čvrstu poziciju u Jugoslaviji pa nije ni čudo što se o njemu dosta priča. Pet je davno prošla stvar. Vic-20 je zaboravljen ali je zato šezdesetcelvroka hit. Komodor 64 je izvanredan za te pare, ozbiljniji (verovatno od spektruma) ,spor u radu sa periferijskom opremom i netačan u računu. On još ima solidnu tastaturu i odvratan dizajn. O novim komodorima se manja priča jer još nisu doprli do Jugoslavije ali se već zna da su ,mocni ,,skupi i zanimljivi. Ulpritka ono što se zna za svaku novu seriju bilo kog proizvođača.

I ostali računari nisu zaboravljeni. O BBC ju se nako vreme dosta pisalo ali je to zamuklo. Ostala je samo priča da je BBC B jako dobar. Epl II je legenda jako je prevaziđen. Makintoš je super stvar i predmet snova tipa ja bih voleo Maka. IBM PC je ozbiljna mašina i ,klasa za sebe. O Atariju se skoro nije ni pričalo ali otkad ja Alari kidnapovao Džeka Tramijala i njegovu svitu na našem tržištu koje nije videlo više od tri atarija smatra se da je nova serija Atarija zanimljiva i sasvim nov kvalitet za Alari.

O domaćim računarima se uopšte i ne priča jer čak i oni koji u računar ne umeju da unesu ni igru sa snobovskim prezirom smatraju da su to kompjuteri za prve koraka u bajziku našlo kao priručni materijal za bejzik.

Dosta je česta pojava privrženosti računaru koji niko nema. Zašto da ne? Tako je dobar ali prosečan ali i jaftin. Amstrad postao maltene legenda pre nego što je bio i stigao do Jugoslavije. Neki zaljubljenici u Hewlitt Packard i dalje tvrde da njegovi računari imaju izvesne dodatne kvalitete malobrojni vlasnici TRS 80 tvrde da je on ipak klasika, poklonici elektrona tvrde da je on nepravedno zanemarivan i tako dalja i tome slično.

Najzahvalniji za blic ocena i aseje iz malog mozga su veoma poznati a veoma nedostupni računari koje samo ješko videti negde u Jugoslaviji. Na njihov račun se može reći bilo šta a što je najlepša svi prisutni hakeri i parahakeri će se odmah složiti sa izrečenim jer ni oni ne znaju o čemu se radi a ne smeju dopustiti da izoslanu iz uskog kruga posvećenih koji su upućani.

Nisu samo računari predmet tako opširnog poznavanja i sve ostalo vezano za računare podleže istom procesu.

Bejzik je spor i primitivan ali i jednostavan i lak. Paskal je kvalitetan i funkcionalan a logo ja ona stvar koju treba više primenjivati mada malo ko zna kako on uopšte i izgleda. I najzad programiranje na mašincu ja prava stvar ali i komplikovana stvar.

ROM je skoro uvek skučen nedovoljan i loše organizovan. Mikrodrajv je spor nepraktičan i promašen. Diskete su praktične i pouzdane dok je disk drajva najbolja imali dva. Zašto? Za ozbiljnu primenu.

Matrični štampač je praktičan štampač sa tapezom je bučan ali ima ,kvalitetan otisak dok je injekt štampač još uvek skup.



Psiho test

pirati i obožavaoci

Specijalni monitori monohromatski ili ne je za određene primene neophodan LCD je nečitljiv iako „praktičan (!?) Grafika visoke rezolucije je obavezna

Video igre su svima dosadne pravazidene „suviše jednostavne ali i „suviše kompliKovane sa kvalitetnom grafikom lošom animacijom i izvanrednim zlaktima Sve je više lvrdnji da su video igre trube i glupost „ali ih igra isti toliki broj ljudi kao i pre

Obavezna laktura za znalce su razno razni časopisi koji pišu o računarima Jadna od poslastica koja se u njima mogu naći su tabela sa brzinama računara Na osnovu tih tabela se iako tvrdi da je neki računar „spor' ili „brz , iako bi trebalo reći da bi računari za primene koje ih znalaze kod dobrog broja jugovlasnika mogli slobodno da budu i dasatak puta sporiji nago što jesu

Gratička tabla plotari svatiosna para i slična stvarčica se ne pominju previše osim u snobovskim krugovima onih koji smatraju da činjenica što imaju dovoljno para za takva dalikatase stoji kao razlog da konverziraju samo o takvim stvarima

Posabna stvar su priča o računarskim časopisima Postoja računarski časopisi kojima tali samo još jedan korek do časopisa za domaćicu i njanu porodicu Takvim časopisima sa smaju i njihovi eopatvani saradnici, pa nija ni čudo da sa za njih priča da au jednostavni i razumljivi Drugi tip časopisa je suviše zanat tehnikom računarskog bitisanja pa tako naočigled svojih čitalace avoluira u tehničku anciklopediju Za takva se kaže da su „suviše kvalitatni (!?) Traća vrsta su oni časopisi koji dobrano filuju svoja strana takstovima preuzatim iz mora svetskih računarskih časopisa Oni važe za pravodiačka časopisa Za jedna se priča da su opasni a za druge da „postaju sve bolji Lako ja primatiti da sam izostavio časopisa za koje bi nako mogao da smatra da našto i vrede To je normalno o takvim časopisima sa malo priča — oni se kupuju

Postoji čitava jedna oblast nabuloznih praganjanja i rasprava koja je vezana za tako apstraktna stvari kao što su operativni sistemi čipovi kompatibilnost svega sa svećim i slično Rasprave o takvim temama imaju tendenciju da budu tako zamršena da polako teže filozofsko-religioznim diskusijama, punim rači kao što su „modulitat apsolutno „apriorno „sublimacija i „rakognitivno i sličnih

Na svu sraću to ja više zabavno nago štatno Tako ća i biti sva dok buda viša onih koji su čuli za računar nago onih koji o računaru našto i znaju

Moglo bi lako da sa dogodi da računari uskoro iskliznu iz nake trenutna „moda pa da čujamo o njima da su dosadni šta je to? „stara tora i obična — „glupost !

Branko Đekovic

Po ugledu na nagradne igre testove i kvizove sa didaktički opravdanim pitanjima i poučnim odgovorima rešili smo da uradimo i više od toga ovo nije samo test kojim ćete proveriti ispravnost svojih stavova već i prilika da se u tom smislu korigujete i nadogradite Sve je zasnovano na strogo naučnim principima i sva sto budete izjavili moći će da sa iskoristi protiv Vas Osim toga možate da učestvujete u nagradnoj i kaznanoj igri

Pažljivo pročitajta pitanja i zaokružite samo jadan od pradloženih odgovora Strogo je zagranjano korišćanja litarature (na primer ranijih brojava „Računara) kao i drugih nedozvoljanih sredstava Na raspolaganju imata 45 minuta

1 Zašto sa domaći kućni računari zovu mišolovka?

a) Sa indignacijom odbijam insinuaciju

b) Na znam tako skupe mišolovka još nisam video

c) Zato što umasto procesora imaju miša

d) Zato što pradstavljaju stupicu za naivne

2 Da li sa pomoću domaćag kućnog računara mogu loviti mišavi?

a) Da mi sve možemo

b) Moraju sa jasno praciziratl ciljavi i zadaci subjaktivnih snaga pa i šira

c) Na ispiati se

d) Ne ni miševi viša nisu tako naivni

3 Yugo-računara traba izvoziti Amarikancima

a) Bio bi to posao stoleca

b) Ima da izvozimo pe šta košta da košta

c) Neka ljudi vida šta mi imamo

d) Naka se i oni malo nasmaju

4. Ako Vam neko traži neki ozbiljniji program, racimo takat procasor, za džabe, šta ćate uraditi?

a) Pošto na možeta tak tako da ga odbijeta reci ćete da program nija predviđan za kasetoton nago samo za disk

b) Uvalićeta mu program sa greškom da on ne bi mogao da koristi program koji vi na znate da koristite

c) Odbićete ga ladno zato što na moža ta da podnesate da još nako ima isto što i Vi

d) Daćate mu tražani program i sve ostale koje imate

5. Imata priliku da preanimita program Vi ćate

a) Tražiti pismano odobranja od stranog proizvođača

b) Sa gnušanjam ćete odbiti i pomisao na tako nešto, all ćeta program ipak presnimiti da zlobnici ne bi mogli da kažu da Vi to na radita zato što na umata

c) Zazmuricete uraditi to u mraku da niko, pa ni Vi ne vidi šta ste to uradili

d) Pitacete da li ima još

6 Ako Vam neko ponudi da Vi sabi preanimite program koji on ima Vi ćete

a) Privesti ga u najbližu stanicu milicije, a program ćeta da prakopirate da biste imali dokazni materijal

b) Išamaraćate ga i oduzeti mu sve programa zato što pokušava da Vas navede na piratiju a i pokušaj se kažnjava

c) Objasnicete mu da Vi sabi na možete dozvoliti takvu niskost nago da Vam on to presnimi

d) Ponudićate mu svoja programa

7 Kako ćeta objasniti ako Vas Komisija za lapitivanja parakla imovina pita odakle Vam toliki programi?

a) Programe sta kupili od stranog proizvođača za davize koja sta pozajmili od MMF daca će da vrata

b) Raci ćete da sta programa pozajmili i da ćate ih uradno vratiti

c) Vi ih samo čuvate da ih nako drugi na presnimi

d) Pitaćate Komisiju da li njima traba naki program

8 Koji ja najprikladniji jezik za školaka računara?

a) Pseudojazik koji ca pradavati psaudoprotasori

b) Trabalo bi ravnopravno upotrabljavati sva jezike naroda i narodnosti

c) Starogrčki

d) Šatrovački

9 Zašto ženama treba zabraniti pristup računarima, a naročito računarskim časopisima?

a) Zato što zageduju okolinu

b) Zato što to nije uobičajano madu narodima jugosavarne Afrika

c) Zato što sa, po pravilu Marica ne razuma u kriv džojstik

d) Zato što je čitalac iz Novog Beograda to tražio

10 Vaša omiljena igra ja

a) Piratbustars (istrabljivači pirata)

b) Brankovo kolo (igra avantura sa jednim životom)

c) Manic Writar (prapisivanja knjiga)

d) Krakari — razbijači zaštite

11 Kako ćate na najbolji način na brzaka da zaradita lavu na račun kompjutera?

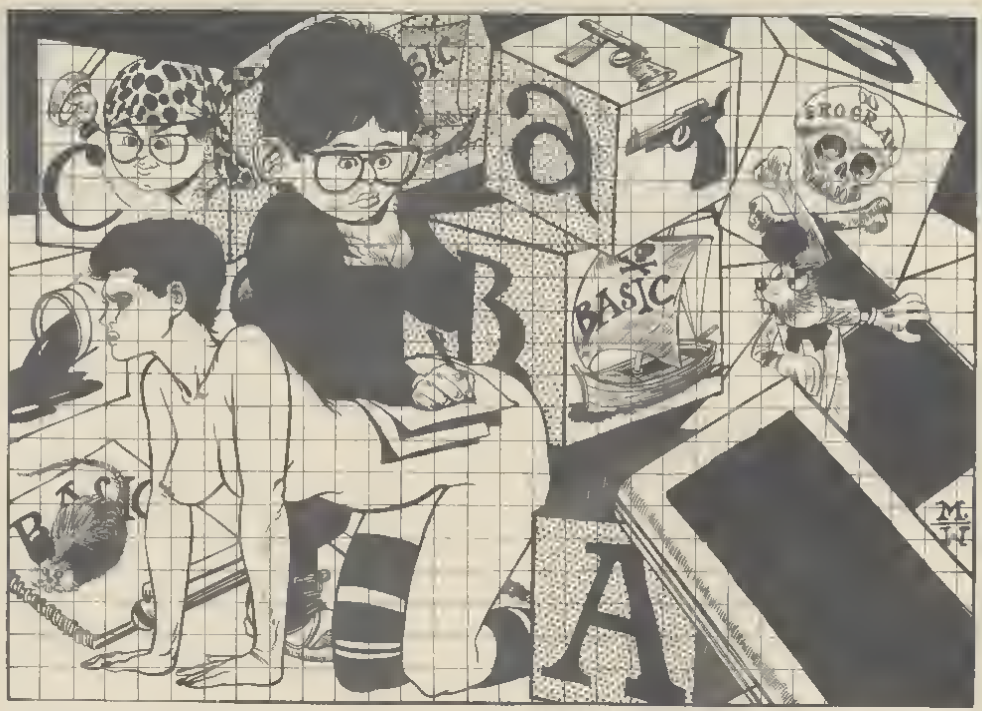
a) Izdaćeta jednu knjigu i dva prijatalja

b) Objaviceta kompjutarski postar se pogrešnim spiskom naredbi

c) Kumovima pešenozima šuracima i ostalom mnogobrojnoj rodbini držaćeta kursave bejzika i prodavati programa 13-5% jeftinija

d) Prodaćate program Englezima

→



12. Naslov Vaše najnovije knjige iz oblasti računarstva biće

a) Kompjuter kao velika nuzda (razmatra probleme društva koje iz informacijske predistorije zakoračuje u takođe informacijski srednji vek)

b) Kako bez muke napisati knjigu o računarima

c) Majstorija na džojstiku

d) Bitak i užilak

13. Gde biata najradija drŽali kuraeve bajzike?

a) U masnoj zajednici

b) U Engleskoj

c) U liftu između prizemlja i osmog sprata

d) U floci

Ako su Vam svi odgovori pod a), Vi ste zdrava zrela i uravnotežena osoba sa visokom moralno-stručnom podobnošću naša radost dika i uzdanica Samo nastavite tako!

Ako ste na sva pitanja odgovorili sa a) i b) Vaši stavovi nisu uvek precizno formulisani ali u Vaše opravdavanja ne bi trebalo sumnjati Traba još da radita na svom usavršavanju i distanciranju od atički štetnih ideja

Ako u odgovorima imata a) b) i c), Vaši su protivrečni i nedosledni ali sta u biti poštena osoba koja luta i kojoj treba pomoći da nade pravi put

Ako pored ostalih odgovora imate i ponako d) Vi ste već zatrovani i očito je da sta pod uticajem loŠeg društva Promanita sradinu i odrecita se prijatelja jer je očigledno da Vas vuku u ponor od svih veci

Ako Vaši odgovori sadrže veći broj d) to je već indikativno U pitanju je teške indokrinacija društveno neprihvatljivim idejama Ipak još ima nade za Vas Potrabno je kompletno ispiranje mozga posle toga nečale moratl da se odreknala prijatelja — oni ce sa odrači Vas

Ako ste na sva pitanja odgovorili sa d) Vi ste naizlačiv slučaj moralnog raspada i najnižih asocijalnih nagona Na samo da niste dovoljno licemarni nago još pokušavata to da kompanzujate svojim cinizmom Vaše pogubno dalovanja je tim opasnija zbog Vašeg nastojanja da i druga navučeta Zbog Vas i sličnih nosilaca pojava zdrave subjaktivna snage moraju povesti anargičnu akciju razolkrivanja i razobličavanja

Nagradna igra za one koji su na sva pitanja odgovorili sa a)a)a)a)a)a)a)a)a)a)

Onaj ko pohvata najveći broj onih koji su na sva pitanja odgovorili sa d) dobiće glavnu nagradu Velikog zlatnog Kerbera, značajno priznanje koje će mu biti dodeljeno uz sve počasti koje dolikuju jednom A-pozitivnom članu naše društvena zajednice

D-nagativni nosioci nagativne pojave koji budu otkrivani i uhvaćeni ili koji se sami prijave morače javno i pred svima da priznaju svoja nedela

Jelena Rupnik

Računari u akciji

baze podataka

Posle obrade teksta, rad sa bazama podataka privlači najširi krug ozbiljno nastrojenih korisnika kućnih računara. Programi ovoga tipa omogućuju da u memoriji kompjutera i na disku držimo velike skupove raznih podataka koje je moguće brzo i efikasno menjati, pretraživati i štampati. Ako je baza podataka integrisana sa tekst procesorom, moći ćemo da je koristimo za pisanje službenih pisama i štampanje spiskova članova nekog kluba ili pretplatnika na neku knjigu i tako bitno pojednostavimo i pojeftinimo organizaciju posla. Za rad sa komercijalnim programima za baze podataka je, jasno, potrebno daleko manje znanja nego za sastavljanja bajzik ili fortran programa koji će obradivati podatke, ali je ipak neophodno upoznati osnovna pojmova i — pažljivo proučiti uputstvo za upotrebu!

Pre nego što se pozabavimo terminima od značaja za rad sa bazama podataka posvetićemo malo vremena jezičkim nedoumicama. Činjenice je da su se kod nas računari (pri tom prvenstveno mislimo na velike kompjuterske sisteme) najpre koristili za obradu podataka što znači da postoje manje ili više dobri prevodi za osnovne engleske termine. S druge strane program za rad sa bazama podataka koji ćete koristili će po svemu sudeći biti uvezen iz inostranstva što znači da će njegovo uputstvo biti pisano na engleskom ako koristimo jedne termine u tekstu a vi pročitate druge termine u uputstvu ovaj napis nema mnogo šansa da bude koristan. Želo smo odlučili da neke od izraza koji su se veoma odomaćili prevedemo a da za druge zadržimo originalna fonetski ispisana imena. Na kraju za one koji imaju problema sa jezikom dajemo mali uporedni rečnik.



Izbor . .

Datoteku (tajl) možemo da zamislimo kao veliki metalni orman sa mnogo fioka kakve gledamo u starim kriminalističkim filmovima ili u novoj još nekompjuterizovanoj Dinastiji. Jedan orman (ili nekoliko ormana koji se nalaze u istoj prostoriji) sadrži fascikle sa podacima o raznim artiklima (u našem primeru sveki prestupnik odnosno svaka naftne bušotine ima po jedan dosije). U kompjuterskoj terminologiji svaki dosije nazivamo **slogom** datoteke (record). Struktura svakog sloga je u većini slučajeva jednaka on ne primer sadrzi nazv bušotine njenu lokeciju datum početka radova na njoj. Informacije o do tog momenta postignutim rezultatima i troškovima ime čoveka koji je zadužen za radove i tako dalje. Svaka od ovih informacija se naziva poljem (field) i sastoji se od slova ili brojeva tako da razlikujemo numerička i alfanumerička polja. Vidimo da je strukture baze podataka hijerarhijska: na najvišem hijerarhijskom nivou je baza podataka (svi podaci koje poseduje firma Denver Karington) ispod nje su pojedine datoteke (npr. datoteka bušotina) koje se sastoje od slogova (podaci o svakoj bušotini) dok se slogovi sastoje od polja a polja od slova i brojeva. Veliki kompjuterski sistemi, a u poslednje vreme i složeniji programi za kucne računare podržavaju i dodatne hijerarhijske nivoe kao što su grupna polja pa čak i razlike u sustini obrade podataka — povecava se jedino njen komfor.

Kapacitet datoteke

Najvažnija karakteristika programa za obradu podataka je koliko kilo/mega/giga bajta informacija može da sazveče. Programi ovoga tipa pisani za kucne racunare se dele na dečije igračke koje rade isključivo sa podacima u memoriji i profesionalne proizvode koji ograničavaju količinu podataka jedino slobodnim prostorom na tlopi (ili hard) disku. Programi koji rade isključivo sa podacima u memoriji se karakterišu brzim pretraživanjima ili sortiranjima i pogodni su za kucnu upotrebu adresar telefonski imenik kucni troškovi spisak softvera i, ne samoj granici njihovih mogucnosti sredivanje manje kucne bibliotеke. Za sve vece poalove program podatke treba da drži ne disku dok se u memoriju upisuje samo jedan njihov manji deo na primer slog koji se trenutno prikazuje ili edituje. Time se jasno znatno gubi na brzini ali je rad i dalje neuporadivo brži od ručnog pretraživanja kartoteke. Da bi se gubici u brzini smanjili primenjuju se razni principi rada sa indeksima i u lančanim listama koji mogu da daju tascinantne rezultate. teko da program za rad sa bazama podataka možemo procenitl jedino ako ga probamo. treba tražiti od prodavca disketu sa podacima koje je skoro puna i proveriti koliko je vremena potrebno za dodavanje novog sloga pronalaženje nekog starog ili sortiranje datoteke.

Brzina sortiranja

Brzina i tleksibilnosi sortiranje su aspeki po kome vecina korisnika sa razlogom rangira programe za baze podataka. Smisao sortiranja kompjuterske baze podataka mozda na prvi pogled nije jasan. dosijei prestupnika u policijskom ormenu moraju da budu sortirani po prezimenu jer bi za coveka bilo praktično nemoguće da ih lista jedan po jedan. Kompjuteru je reči ce neko svejedno da li ce pretraživati sortiranu ili nesortiranu datoteku. čilace je slog po slog dok ne pronede onaj koji mu je potreban. Nije medutimm bas tako najčesce se od računara traži da slog pronede po nekom fiksnom polju. ne primer po imenu i prezimenu. Ako je datoteka sortirane po ovom polju. računar će moći da primeni binarno pretraživanje i daleko brže pronade ono sto mu je potrebno. Ukoliko se datoteka pretražuje po nekim drugim kriterijumima od čitanja svakog posebnog sloga se teško može pobeci.

Loši algoritmi za sortiranje imaju red složenosti n što znači de će za sortirenje 200 slogova biti potrebno četiri puta više vremena nego za sortiranje 100 slogova. Dobri algoritmi za sortiranje imaju red slozenosti n log n što znači da je za sortiranje 200 slogove potrebno 2.3 puta vise vremena nego za sortiranje 100 slogova razlika izmedu ovih veličina postaje daleko značajnija kada broj slogova reste. Osim toga daleko će brže sortirati program koji na disku drzi sadržaje slogove a u memoriji

→

teblicu sa njihovim redosledom — slogovi se tada tiziški ne pomeraju već se samo zamenjuju redovi u teblici Sve u svemu, rezlike u brzini reda programa ze sortirenje mogu da budu neverovetno velike i njihovo poredenje je opravden kriterijum za rangirenje sistema za baze podeteke

Tehnike pretraživanja

Pretraživenje je sledeća radnje koju program za rad ae bazame podetaka uvek podrževa Brzine pretraživenje obično nije kritična program kome sa zadeju neki složenl uslovi po kojima treba da pronade slog će morati da čita detoteku od početka prema kraju Kod pretraživenje se međutim ceni tlekselbilnost treba omogućiti korisniku da zadeje složene uslove po kojima će program vršiti pretraživanje Možemo, ne primer de zahtevamo da se pronedu svi prestupnici koji žive na Zvezderi i koji su osudivani za aitne krade a nisu sa nslazili u zetvoru na den 1 marte 1985 godine Od programe za pretraživanje koma je postevljen ovekav zahtev očekujemo odredenu dozu inteligencije on treba da zna da neko ko je bio u zetvoru od 1 januera do 15 aprile nije mogao de bude na Zvezderi prvog marta i de je u nekom još komplikovenijem slučeju Vranjska ulice na Zvezderi e Čolak Antina nije Posledniji primer pokazuje da ne velikim sistemime pa i ne manjim računarime koji su povezani sa herd diskom ili kompjuterskom mrežom program ze bazu podeteka može da konsultuje još neke banke podataka (u našem alučaju, kompjuterski plan grede) osim detoteke koju pretražuje Ukoliko želite da znete keko ce sve to izgledati jednoga dane preporučujemo vam neučno-fantastični romen Frederika Pola Kapija 2 '

Želeći de ubrzeju pretraživenje autori programe ze baze podeteke često ae odlučuju da oslabe njegove kerekteristike nije malo ček i vrlo skupih programa koji pri pretraživenju uzimaju u obzir samo prvih šest ili deset slove (brojeva) u nekom polju Ovakva solucije moze de bude dobre u nekim primename dok u drugim predstevlje veliko ograničenje Medutim neki programi ovoga tipa omogucevaju da se ze svaki slog definiše primerni ključ to jest polje od četiri ili pet slova koje ce računar držati u memoriji zajedno se tebelom sa indeksime slogove Ukolilko je u to polje upisane skraćenice koja kerakteriše (ne)dele nekog prestupnika računar ce praktično trenutno moći de pronede sve registrovane prodavce nerkotika ze nekoliko sekundi dok bi učitavenje svih slogova sa diska odnslo nekoliko časove

Editovanje slogova

Četvrte keraktenistike programa ze baze podetaka koju emetremo bitnom je mogućnost editovanje postojećih elogove Jesno je da sveki program ovoga tipe omogucave de se podaci u nekom slogu izmene ali pravi problem nsetaje tek onde kede editovanjem slog postane bitno duži nego što je bio Keko su slogovi guato pekoveni na disketi ze ovo produženje neće biti meste pa ce slog biti obrisan i prepiaan na neko drugo slobodno mesto čime će se u bazi podateke pojaviti rupe Dober program ce keda disketa bude pune izvršiti takozveni , tresh colecting i kompektovati postojece slogove čime će se ne kraju datoteke atvoriti veliki sloboden prostor Osrednji program će ponuditi korisniku bajzlk rutinu koje prepiauje koristen sadržej diske sa praznim proatorime na tek formatirani disk dok će loš program jednostavno ignorisati ovaj problem

Organizacija programa

Može se dosta rspravljeti o tome de li su bolji oni tekst procesori organizovani po menijime ili tekst procesori u kojime treba

Lozinke

Vlasnicima kucnih računara ne treba mnogo govoriti o zaštićenim programima — dobro im je poznato da su softverske firme smislile mali milion metoda koji kupcu omogućavaju da nesmetano koristi program, ali ga sprečavaju da ga kopira i dalje distribuira Poznato je isto tako, da ni jedan od tih metoda nije stoprocentno siguran

Zaštita podataka je sasvim drukčija stvar njen cilj je sprečavanje neovlašćenih lica da čitaju poverljive informacije Disketu sa zaštićenim podacima biste dakle nesmetano mogli da poverite bilo kome na čuvanje, on bi bio u stanju da napravi neograničen broj kopija, ali i dalje ne bi bio u stanju da pročita podatke koji su upisani na njoj Da stvar bude još lapša neovlašćeno lice (ili bad guy kako ga srkovito zovu u američkoj literaturi) nema načina da na smislen način promeni podatke na disketi on jseno, može da ih uništi, ali ne i da modifikuje neke od njih i teko sebi pribevi bilo kekvu korist.

Zenimljivo je da se ne može postići atoprocentne zaštite podataka štaviše, ček i početnik može bez mnogo probleme da aeatavi bajzlk program koji će zaštititi podetke u bito kojoi detoteci

Kako taj program redi? Najpre treba de sastevimo proceduru koja, na osnovu broja koji zedamo generiše sekvencu pseudoslučejnih brojeva izmedu 0 i 255 pri čemu je vrlo bitno da isti početni broj uvek generiše istu sekvencu Jedna od najjednostevnijih procedure koje ispunjavaju taj uslov je

```
1000 DEFPROCstbr
1010 B=EXP (SEED+3 1415927)
1020 SEED=B-INT (B)
1030 SLBR=INT (B*256)
1040 ENDPROC
```

Program koji šifruje detoteku glasi otprilika ovako

```
10 INPUT  Početne lozinka  SEED
20 INPUT  ,Ime izvorne datoteke  DI$
30 INPUT  ,Ime objaktne detoteke  D2$
40 X=OPENIN D1$
50 Y=OPENOUT D2$
60 REPEAT
70 A=BGET#X
80 PROCstbr
90 A=A+SLBR IF A>255 THEN A=A-256
100 BPUT#Y A
110 UNTIL EOF#X
120 CLOSE#X CLOSE#Y
130 DELETE D1$
```

Princip reda progreme je sesvim jednosteven čita se prvi bajt ulazne detotake uvećave za etučajen broj, e zatim se razultat upisuje u izlaznu datoteku Kada ne kraju ulazna detoteke buda obrisana, sedržej izlazne detoteke će biti besmislen ze evekoga osim onoga ko zne lozinku On će kada bude želso da koristi podetke, izvršiti program koji se od gornjeg razlikuje samo u jednoj liniji

```
90 A=A-SLBR IF A>0 THEN A=A+256
```

Startovenjem ovoge progreme i unošenjem lozinke (istog decimelnog broja koji je unesen keda je detoteka šifrovane) dobiće se početna detoteke dok će ,zaključena verzija biti obrisena

Programi ze baze podetaka obično sedrže generetora pseudoslučejnih brojeve koji su dalako bolji od onoga koji smo mi deli i koji omogućavaju da početna lozinke bude rač e ne broj ali je princip njihovog delovanje potpuno lati lozinke nije zapisena nigde na diaku i ne postoji nikakav način da nako ko ja na zne u nekom zamislivom vramenskom intarvelu od par hiljada godine pročita ,zakljućana podatka

Nevolje neataja kede vlasnik podataka zaboravi lozinku — njegov višemesečni rad se začas pratvare u gomilu besmislenih podeteka Osim toga, ukoliko koristita naki komercijalni program i zaštitite svoja podetka lozinkom verovatno načete biti u stanju da ih čitate i obradujete primanom naših bajzlk rutina koja čete, uz pomoć ‚Utility dieka' samostalno napisati Čini se da je zbog svega ovoga, daleko lakše i bezopasnije zaključavati diskete nego podatke na njima. Na recita da vas nismo upozorili!

kucati komende ali je takva dileme kod baze podataka rešene — svi vredni programi ovoga tipa pisani za kucne računare su organizovani po manijima jer se pokazalo de samo tako korisnik može da iskoristi eve potencijete programa Vazno je de osim menije program ime i neku vrstu HELP mode u kome će korisnik dobiti obevaštenje o funkcijama koje su pridružene soft lesterima i eventualnom značenju slova kade je zajedno se nakim od njih pritisnute dirka CTRL Ovakve organizecija programe može medutim de dovede do drugog problema za obevljenje neke sasvim obična funkcije potrebno je proci kroz nakoliko nivoe menija i prema tome pritisnuti desetak tastera Samo sa u praksi može pokezati da li su autori nakog programa napravili dobar kompromis izmedu brzine rada sa programom i jednostavnosti njegova upotrebe

Fleksibilnost programe je poslednja kerektenatike o kojoj imemo nemeru da govorimo Jasno je de autori programe ze rad sa bazame podeteka ne mogu de predvida sve eitnica koje će nem zetrebati Tim pre oni ne mogu de predvide mogućnosti tekst procesora koji koristimo karakteristike neŠeg printara i tipove podetaka koje smo ne neki drugi način upisall ne disk Zato je vrlo zgodno eko se korisniku programa za baze podeteka koji poznaje bejzik omogući da uz umeren trud sastavi programe koji će prstupati bezi podataka i veditt podtke iz nje ili programe koji ča pripremati podatke na način koji baze podeteka može da prihveti Zato je zgodno kupiti program za rad se bazama podataka uz koji sa dobija i „utility diak ' sa skupom bejzik i mašinskih potprograme

. . . *i kreiranje datoteke*

Pošto sledeći saveta iz prathodnog poglavlja i proučivši prikaze programa za baza podatake iz stranih časopisa i sledećeg broja računare odabaramo program koji nem odgovara treba de uložimo malo vramane u privikavanja na njega Baz obaira na činjenicu da ste računar možda kupili baš zbog rada sa bazeme podataka i da vem je važno de što brže počnata da koristita njegove usluga, savladajta nestrpljenja i sastavite jednu vrlo jednostavnu bazu podataka koja ča vam omogućiti da vežbate Nama naravno praprake de počnete sa kreiranjam baza podataka koja vam je zaista potrabne i da se učita na njoj ali traba de sa pominte sa tim de cate ovu bazu podetake vrlo brzo morati da uništite i da zepočnete posao od početka — ona jednostavno nace biti zasnovana dovoljno ileksibilno Zbog toga vam predlažemo da zajedno sa nama pokušeta da kreirete datoteku se imanima adrasama i drugim podacima o vašim poznanicime u koju cete moći da upisujeta i napomana o tome šta koma dugujeta i šta vam ko duguja

Zepočinjanje baza podataka je baz eumnje najneprijatniji deo posla traba da se odlučite za vatičinu sloga i za polja koje ća ga sačinjavati Značaj ova operacija verovatno ne treba posebno objašnjavati — stog traba da buda dovoljno velik da omogući smeštanje svih podataka koji su vam potrabni i svih podataka koji ća vam biti potrebni u buducnosti a da istovremeno bude dovoljno mali de ne zeuzma preveliki deo diska Ze našu datoteku ce nem biti potrabna polja čije su dimanzije prikazene na slici 1

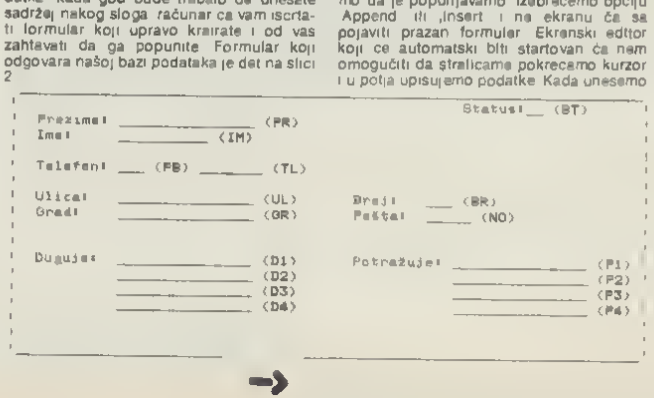
```
Prezime:          15 (A)
Ime:              10 (A)
Telefon:          10 (N)
Ulica:            15 (A)
Broj:              2 (N)
Grad:             15 (A)
Poštanski broj:    5 (N)
Status:            2 (A
Duguje:           45 (A)
Potražuje         45 (A)

A: alfanumeričko polje
N: numeričko polje
```

Pošto na papiru ispleniramo slog stertovecemo program koji će nam predsteviti glavni meni u kome je jedna od opcija svakeko kreirenje nove detoteke Izabračemo tu opciju (verovatno će biti potrebno da tekvu odluku potvrdimo jednim Yes obzirom de njano izvršavanje možde uništeve deo sadržaja diskete) Računar će proveriti de li je diskata koju smo pripremili korektno formatirana i dovoljno prazne (neki programi ovoge tipa zahtavaju da za svaku bazu podateke odvojimo po jednu disketu) i ako je sve u radu zehtevati da otkucamo ime baze podataka Ovo ime nije neročito bitno sa funkcionelnog aspekta ali ce nam pomoci de docnija pronademo diskatu se podecime koji eu nam potrebni Slede pitenja o detumu kreiranja i nekim drugim podacime medu koje kod iola boljih programa spada i pitanja o potrabi za šticenjem beze podetaka lozinkom koja ca onamogućiti ostalim ukućanime da zaviruju u imene vaših prijatelja (i prijatalica)

Definisanje sloga

Sledaća faza ja etartovanja ekranskog edltore koji trabe da vam omogući de definišata izgled sloga (u siromašnijoj verzji ća vam jednostavno biti postavljene pitanje o svakom polju) Konstruisanja sloge možeta da zamislita kao pripremanja matrica u koju cete docnije ubacivati podatke kada god bude trabalo de unesete sadržej nakog sloga računar ca vam iscrtati formular koji upravo kreirate i od vas zahtevati da ga popunite Formular koji odgovara našoj bazi podataka je det na slici 2





Na slici vidimo da je svako polje dobito po neko ime ispred koga je upisana skracenica Ime je tskljućivo estetska prirode i biće prikazivano zajedno se formutarom — računar ga nače upisivati ne disk zajedno se podacima niti čamo moci de ga referenciramo pri sortirenju ili pretraživenju datoteke Skracanica u zagradi tekozvani tag nema (osobina koju imaju samo bolji progremi), upisuje se zajedno se podacime na disk i može de sa koristi za referenciranje polje „Teg nema' ja ne prvi pogled nepotreban luksuz kada vec postoja obična imene Ipak na tej se način značajno šteđi memorije a istovremeno povećava komfor rade jer pri unošenju podataka ne morata da razmišljate o toma da li je GR polje za neziv grada ili oznake grupe u koju ste svrstali vaseg prijatelja Dobar program za rad se datotekama ce omogućiti definisanja primarnog ključa za sortiranje u našem slučaju to je polje STATUS u koje možete da upisujata dvoslovnu skraćenicu koja ce bliže odradivalt osobu na koju se slog odnosi (npr PR za prijatelja RO za rodake KL za kotega CO za vlasnika kompjutara ista firma PI za entuzijasta od kojih kupujate programe i tako dalja) Ovo polje ce ako ge dobro upotrabita omogućiti da višestruko ubrzate buduca pretraživanja

Popunjavanje formulara

Pošto je prakriveč (ovarlay) sloga definisan računar će nas vratiti u glavni meni — datoteka je kreirana i možamo da počnemo da je popunjavamo Izebracemo opciju Append ili ‚Insert i na ekranu ča se pojaviti prazan formular Ekranski edttor koji ce automatski biti startovan ća nem omogućiti da strelicame pokrecamo kurzor i u potja upisujemo podatke Kada unesemo

REČNIK ŽARGONA

AC...S — način pristupa datoteci
...IC FIELD — polje u koje se smeštaju ključne vrednosti
...FIELD — polje u koje se ne stavlja slova i (eventualno) i brojevi
...REND R...RD — menjati neka polja u slogu
...PEND — dodavanje slogova na kraj datoteke
...ATE ARIABL... — promenljiva (ponekad interna) koja ukazuje na sledeći slog datoteke pri sekvencijalnom pristupu
...CKUP — formiranje kopije datoteke koja će biti korišćena u slučaju kvara na mediju na kome se datoteka čuva
...SKOK — skup slogova koji se zbog bržeg rada zajedno učitavaju u bafer
...WSE — pregledanje datoteke ili podskupa slog po slog
...NCODE CONTROL — deo ili ceo se na kraju sloga nalaze CR CR i LF i blanko
...ARACTER — slovo broj ili specijalni znak
...ARACTER SET — sva slova i znakovi kojima računar i štampač raspolažu
...SE — zatvaranje datoteke
...SE BASE — baze podataka ili program za rad sa bazama podataka
...FAULT — vrednosti ili opcija koje se podrazumevaju ako se ništa ne navede
...LETE — brisanje
...ECT ACCESS — način pristupa kod koga se može direktno čitati bilo koji slog
...TING — promena nekih podataka
...CAPE SEKVENCA — sekvence kodova koja se šalje printeru. Počinje sa 27
...TEND SIZE — uvećanje prostora na disku koji je dodeljen datoteci
...LD — polje deo sloga
...E — datoteke skup slogova
...ISH — regularan završetak — promene se upisuju u datoteku
...ED RECORD — svi slogovi imaju jednaku dužinu
...RMATTED FILE — datoteke koja sadrži ASCII tekst
...E — naredba kojom se od računara traže podaci o drugim naredbama
...XED FILE — datoteke u kojoj se slogovi pronalaze prema sadržaju jednog polja — ključa
...UT FORM — matrica prema kojoj se unose podaci
...ED ACCESS — način pristupa indeksnim datotekama
...AIL-MERGING — pripremanje pisama i etikala za kovertiranje na osnovu sadržaja datoteke
...NUS — meni pomoću koga korisnik kontroliše rad programa
...DE — mešanje datoteka tj formiranje nove datoteke od više postojećih
...UMERIC FIELD — polje koje sadrži brojni podatak (obično će obrojan)
...EN — otvaranje datoteke
...SWORD — lozinka koja sprečava neovlašćeni pristup podacima
...MARY KEY — polje po kome sa datoteke sortira
...NTER DRIVER — program pomoću koga se podaci prenosa na štampač
...NDOM FILE — datoteke u kojoj u svakom trenutku može direktno da se pristupa bilo kom slogu
...D ONLY FILE — datoteka koje sme samo da se čita
...CORD — slog skup podataka o nekoj osobi/predmetu
...VIEW — brzo pregledanje sadržaja datoteke
...CONDARY KEY — polje po kome se sortiraju slogovi se istim primarnim ključem
...QUENTIAL ACCESS — čitanje (ili upis) datoteke slog po slog
...QUENTIAL FILE — datoteka u kojoj se slogovima mora pristupati redom
...ORT — rešanje slogova u rastući ili opadajući niz
...TATUS LINE — prve ili poslednje linija ekrana u kojoj vid mo obaveštenje o nome što program u nekom trenutku radi
...UBSET — odreden broj slogova izdvojan iz datoteke
...UM NAME — stvarno ime polja
...TLE — naslov ili ime datoteke
...UNFORMATTED FILE — datoteka koje sadrži brojeve u internom formatu
...PDATE FILE — ažuriranje datoteku
...ER NAME — ime korisnika obično se navodi pre lozinke
...RIABLE RECORD — slogovi su promenljive dužine
...RITE FILE — datoteka kojoj (na velikom sistemu) istovremeno pristupa više korisnika.

čitav slog pritisnućemo neki od „sottlastera, dobiti novi prazan formular i ponoviti čitavu operaciju. Pri unošenju podataka treba biti veoma pažljiv jer ćemo svaku grešku docnije skupno plaiti ako umesto PERA PERIĆ otkucamo PERA PERUĆ program za pretraživanja neće moći da nade podatka o ovom našem prijatelju ieko smo sigurni da smo te podatke otkucali. Za ovu grešku smo, na naki način sami krivi pa nememo mnogo razloga da se ljutimo ne progrem koji koristimo. Neki programi ze rad sa bažama podataka međutim prava razliku izmedu velikih i malih slova pa nećamo moći da pronađemo slog koji sa odnosi na PERU PERIĆA ako je u njega upisano Pera Perić. Stvar će biti još lepša ako podatke o polovini poznanika kucamo danas pri čemu pišemo prvo prezime pa ime a ostatak dopišemo sutra i to nerevno obrnutim redom!

Sortiranje i pretraživanje

Pošto unesemo sve podatke vratićemo se u glevni meni i zahtevati sortiranje. Računar će tražiti da navedamo ime polje koje se zova primarni ključ. e zatim imena potje sa sekundernim ključevima (primarno polja za sortiranja bi po svoj prilici trabalo da bude prezime, sekundarno ime a sledeća mesto stanovanja). Datoteka će, zatim biti sortirana po primarnom ključu (ako se radi o numeričkom polju manji brojevi će se nalaziti ispred većih a ako se radi o attanumeričkom polju sortiranja će sa obaviti po abecedi) a zatim će svi slogovi koji imaju jednaki primarni ključ biti sortirani po sekundarnom ključu a oni sa istim primarnim i sekundarnim ključem po sledecam specificiranom polju. Zavisno od broja slogova sortiranja je operacija koja najčešće ume da potraje.

Kada nam docnija budu zatrabali podaci o nekom od prijatelja izabracemo opciju Ssarch), na način koji je opisan u okviru uputstva za upotrabu programa koji koristimo zadati uslove prema kojima se pretraživanje vrsi. Pri tom će svakako biti omogućano korišćenja džoker znakove o kojima smo govorili u napisu diskatna veza u Računarima 5. Ukratko,povisilica (number sign „taraba , #) zamenjuje bilo koja slovo a zvezdica bilo koju grupu slova tako da će SEARCH MI*IĆ pronaći sva MIIĆE (ako takvo prezima postoji), MIŠIĆE, MILIĆE i MIRIĆE dok će SARCH MI*IĆ uz to pronaći i MtLJKOVIĆE MILANOVIĆE i slične. Dobri programi za baze podataka omogućavaju deteko tlaksibilniju (i semim tim složenlju) upotrabu džoker znakova koja vam verovatno neće nikada biti potrabna.

Umesto jednostavnog pretraživanja kvalitetni programi omogućavaju kraranje takozvanih subseta (subsat). Ako očekujemo da će računar pronaći mnogo slogova koji ispunjavaju data uslove verovatno nam neca biti zgodno da ih samo na kratko pogledamo — ukoliko propustimo slog koji smo tražili moracamo da ponovimo čitavu operaciju i tako izgubimo doata vremena. Zato od računara možemo da zahtevamo da kraira odvojenu datoteku (subsat) koja sadrži samo slogove osnovne datoteke koji zadovoljavaju neka uslova. Docnija camo moći brzo da pragledamo ta slogova i pronadamo onaj koji nam je potreban. Koriat od subseta je i subjektivna — ako očekujemo da neko pretraživanja dugo traja možamo da ga startujemo pa da se onda bavimo nakim drugim poslom. Ukolilko računar prikazuje slog čim ga pronađe taj posao ćemo iz časa u čas morati da prekidamo, dok za krairanja čitavog subsata nija potrabna nikakva intervencija — posvetićemo pažnju računaru tek kada bude u stanju da nam brzo prikaže podatke koje smo tražili.

Programski paketi

Program za rad sa bazama podataka postaje zaista koristan tek kada ga poveže mo sa drugim sottverom. U taj sottver spada pre svega taksi procesor traba da proućimo mogućnosti pranošanja sadržaja nakih slogova u tajl sa tekstom koji ćamo docnije ispisati. Neki programi su opremljena printer drajvarima koji omogućavaju direktno pranošenja podalake na papir ali je fiakabilnost takvih drajvara retko dovoljna ćak i za najjednostavnija poslovna primena. Zato ćete na tržištu neći razna programe koji se zovu Mail list (ispisivanje etiketa ze pisma uz korišćenja imana i adresa iz baze podateka) FieldCalc (izračunavanja sredniih ili prosačnih vrednosti sadržaja nekih polja), Consbase i slično. Ovakva programa ćete na način koji smo vec opisali s vremena na vreme morati da dopunjavate sottvarom iz kućne radinosti.

Dejan Ristanovic

Programi koje treba imati

ne daj se, ines!

Među programskom opremom koje ličnim računarima daje pravu vrednost, porad programskih jezika u prvi red se ubrajaju tekst procesori, programi za obradu podataka, vođenje knjigovodstva i uopšte, uslužni programi. Neki od njih su odigrali pionirsku ulogu u razvoju softvera, recimo VISICALC, drugi su postali veliki hitovi — WORDSTAR, dBASE II — a trenutno su na ceni kompletni programski paketi kao LOTUS 1-2-3, SYMPHONY, ili FRAMEWORK. Većina tih programa je namenjena „apiu", IBM-u, ili sličnim računarima, dok sa za jaftina mašine poput „spektruma" uglavnom prodaju samo igre. Ipak, i za običnog, smrtnog YU hakera kome je IBM san, a „spektrum" realnost, od skora postoji rašanje- INES — program za obradu takata i podataka, dalo jednog našeg programara iz Slovanija, Primoža Jakopina

Tekst procesor je sigurno najvažniji deo INES-a — od padeset naredbi koliko poznaja ovaj program skoro četrdesat je namanjano upravo pisanju na računaru Da bi sa obezbedio konforan rad na akranu sa prikazuja takst u formatu 22 reda puta 64 znaka Naravno to spektrumova razolu cija vrlo taško podnosi ati drugog rešenja sigurno nija bilo Takoda u cilju prijatnijeg rada slova su bala na tamnoj pozadini a tekst se na ekranu polako pomara baz naglih skokova od kojih posla dužag rada zaboli glava

Lako oblikovanja teksta brzo ispravlja nje grašaka, mogućnost da sa vrše razne izmana i jednom rečju alastičnost u radu ja ono što se očakuje od svakog tekst procesora, i INES sigurno nace iznavariti očekivanja Međutim- INES pruža i naka pogodnosti koje slični inostrani programi, recimo slavni TASWORD tl, ne pružaju

Pra svaga- lu su za nas vrlo važna slova č ć š ž i d baz kojih obrada teksta gubi svaki smisao Šta više autor sa pobrinuo da omogući INES-u da ta slova i odštampa a na samo prikaže na akranu Pogodnost važi samo za vlasnika štampača iz serija EPSON FX DELTA i STAR GEMINI ali to su, po svoj prilici, i najrasprostranjanij štampači kod nas na računajući Sinklerovu tarmičku

Inesov rečnik

		I	insert character	s	substitute
		i		t	setup
		J	insert line	T	multiple records
a	adjust to right margin	J	Jump	u	unadjust
A	Append	k	kill to end of line insert	U	Upper to lower case
b	beginning screen	K	rank	v	verify
B	Bytes left	l	last screen	V	dissolve
c	change	m	mark	w	delete word
C	Copy	M	Move	W	Where
d	delete character	n	next screen	x	exchange
D	Delete line	N	mailing list	X	display pixels
e	ending screen	o	order	y	center line
E	Exit	O	turn around	z	zero and insert
f	find	p	pick	Z	set end-of file
F		P	Print	/	change places
g	Frequencies	q	sequance numbers		upper to lower case
G	lengths	Q	Query		duplicate file FILE
h	help	r	restore		join files
H	Help	R	Remove		

Zatim postoji dosta korisna mogucnost štampanja cirkularnih pisama Naime kod ovakvih pisama glavni sadržaj uvak ostaja isti ali sa zaglavlja manja od primerka do primarka INES ceo posao obavlja automatski

Ono sto ja medutim sigurno najinteresantnije i daleko najkorisnije je mogucnost da se razne slike i tabele uključe u takst Za slike se morate sami pobrinuti pomocu programa MELBOURNE DRAW na primar ali zato INES moza da stvori sve tabele koje mogu da zatrebaju Program za obradu podataka je naime sastavni deo INES-a a rezultati njagovog rada se lako i brzo mogu preneti u takst procesor Naredbe koje manipulišu podacima nisu naročito brojna ali su zato izuzetno mocna Za treni oka može se ispremeštati cala datoteka, po odredanom kriterijumu odstraniti sve što na treba sortirati ostatak dodati još ponašto formirati pragladnu labalu i rezultat odštampati Ovakva mogućnosti kombinovane sa lakšoj procesorom širom otvaraju vrata INES-u za sve primene

Valiki problam kod ovakva vrste programa je raspoloživa memorija INES nije kratak program ali je ipak za korisnika ostalo prohvatljivih 21K što iznosi dobrih deset kucanih strana Vrlo važna je i kompatibilnost sa Sinklerovim mikrodrajvom To olvara nove horizonte i olaksava rad Obrada podataka sa običnim kasetofonom je teško zamisliva i vrlo mukotrpna

Zajedno se INES om dobijaju se jos tri posebna programa — DENIS SORTI i UTE

DENIS služi za unosenje podataka u skladu sa odradanim oblikom formulara i za istovremenu kontrolu ispravnosti unatih podataka

SORTI je namenjan sortiranju vrlo valikih datoteka (do 37K) koje ne mogu da stanu u INES

UTE ja najzanimljiviji od sva tri dodatna programa Omogucava da se u INES prane su datotake formirane pomocu nekog drugog programa To mogu biti izvorni programi iz davpaka ili preskola programi u bejziku, takstovi napisani pomocu TASWORD-a delovi RAM-a ROM-a ili slika

Uputstvo za upotrabu od 68 strana koje prati INES ja vrlo opširno i pruža mnoštvo primara Inaća tNES sa po cani od 1500 dinara može naručiti prako Mladinska Knjiga iz Ljubljana

Kako to rade Englezi

TASWORD I (TASMAN SOFTWARE Teksprocesor sa mogućnošću obrade samo 32 karaktera Ekran je ograničenje ga čim potpuno neskladišta za tekst zbog prilagodenu

TASWORD II (TASMAN SOFTWARE) Prvi i pred INES a verovatno ed najrazbijanji program za obradu teksta namenjen spektrumu Može da radi sa 32 i 64 karaktera u redu Raspolaze mnoštvom vrlo mocnih naredbi i podržava EPSON ovi štampace iz serija FX Vrlo je popularan i na zapad r ima nas mada naravno ne raspolaze na im slovima č ć š ž Može se vrlo lako prilagodit za rad sa mikrodrajvom

YU FILE (PSION) Jedan od prvih programa za obradu podataka namenjen spektrumu sa 16 ili 48K RAM a Ograničen mogucnosti Ne predstlav a naročit dobar zbor Nije kompatibilan sa mikrodrajvom

SMALL BUSINESS ACCOUNTS (WILLDEN) Program namenjen malim firmama maloj privredi") za lakše vođenje knjigovodstva Okrenut e vrlo uskom području primene Nije slekao naročito popularan si Nije kompatibilan sa mikrodrajvom

MASTER FILE (CAMPBELL SYSTEMS) Je dan od snažnih programa za obradu podataka Ima mogucnost da pokaže 51 znak u redu što predstavlje lep kompromis zmedu velikog broja slova u redu cilji vosti Lako se koristi zahvalu je tome što se upravlja nje vrši preko menija Raspolaze dobrim i vrlo brzim komanda a Moze da obraduja do 32k podataka Kompatibilan je sa mikrodrajvom

YU CALC (PSION) omogucava razna izračunavanja sa podacima sredenim u obliku matrice Moze da radi na spektrumu sa 16K RAM a Nije se nesto posebno prosla vio Nije kompatibilan sa mikrodrajvom

Vladimir Kostić

Računari u poslovnoj primeni
mali računari u velikoj privredi



Na ni toliko mali kao što izgleda. Čak i za najmanji kućni računar u svakoj rednoj organizaciji može se naći čitav niz praktičnih poslova

Računarski računar namenjeni obradi poslovnih podataka već dvadesetak godina obavljaju u našim radnim organizacijama. Njihova centralna memorija se meri megabajtima a kapaciteti diskova gigabajtima magnetne trake imaju velike brzine prenosa kao i gustine upisa a modemi omogućuju da sa korišćenjem telefonskih linija formira čitava mreža računara. Zar u takvim uslovima naš kućni ljubimac stvarno ima šta da traži? I ta kako! Zašto? Postoje uglavnom tri razloga

• Veliki broj računarskih centara nije kadrovski osposobljen da se bavi svim poslovima koji su od interesa za radnu organizaciju. Pri tome se uglavnom misli na nadovoljnu usklađenost kadrovskih i tehničkih resursa u računskom centru a mnogo manje na stručnu osposobljenost radnika. Većina radnih organizacija koristi svoje računare za obavljanja poslova vezanih za praćenja finansijskih promena (kupci dobavljači fakturisanje lični dohodci osnovna sredstva i slično). Kako je ova oblast vezana za praćenja zakonskih propisa koji se veoma često menjaju (kontni plan stope revalorizacija pravilnici o raspodeli) a skopčani su i sa propisanim rokovima radnici u računskom centru prilično ako ne i čitav dan radnog vremena koriste za tekuće održavanje programa. Na taj način im ostaje malo slobodnog vremena da ga posvete nekim drugim oblastima primene koje su možda i mnogo korisnija za radnu organizaciju ali nisu bile praćene na računaru i nisu bile razvijene jer su zakonski propisani rokovi nametali prioritet na prethodno navedene poslove.

• Izvestan broj radnih organizacija s obzirom na na male teškoće koje prate zamenu kompjuterskih sistema (uključujući tu zakonske propise finansijska problema pa konačno i problema pranosa razvijenih aplikacija na novi sistem) do te mere je opterećito svoju opremu da je praktično nemoguće postaviti na sistem naku aplikaciju.

• Uvek postoje poslovi koji su od interesa samo za neki manji deo radne organizacije. Ovakvih poslova ima jako mnogo — mogu biti čak i vezani za sam proces proizvodnje (kontrola alata mernih instrumenata preventivno održavanje oprema i slično) ili van procesa proizvodnja (kadrovska evidencija normativna akta biblioteka itd) — ali verovatno nikada neće ni biti poslavljeni na veliki računski sitem.

Veliki i mali

Da bismo na naki način uporadili veliki računar za obradu podataka sa kucnim računarom analizirajmo pojedine delove velikog sistema — centralni procesor centralnu memoriju jedinice sa direktnim pristupom magnetna traka štampač i komunikaciona vaza — kao i tri posebna paramaitra

— programske jazika i pomocne programa brzinu rada i tačnost. Pri tom nikako na traba zaboraviti i odnos cena koji posloji izmedu ovih sistama (primara radi samo terminali za veliki sistem danas kosta preko 100 miliona starih dinara)

Centralni procesor

Ako izuzmemo sistema nekih proizvoda ča računara za obradu podataka koji su danas prisutni na našem tržištu (IBM UNIVAC ICL kao i naki modeli NCR DELTA HONEYWELL, ERA) vecina računara koji su instalinrani poslednjih godina kod nas su izgradani oko somobitnih mikroprocesora (NCR serija 90 HONEYWELL sistem 6 ERA 20 i 60 manji sistam DELTA) — znači oko istih ili sličnih mikroprocesora kao i kućni računar. Repartoar mašinskih instrukcija je praktično isti što ima veoma velikih posledica na samu organizaciju čitavog računara.

Od ovog broja dobili smo novog saradnika koji će voditi rubriku za poslovnu primenu kucnih računara. To je MIHAILO KARAPANDŽIĆ dipl. ing. elektrotehnike koji na poslovima AOP-a radi vec više od 20 godina. Odrastao je na velikim sistemima. Radio je prvo u Zavodu za primenu elektronicnih računara iz Zagreba koja je zastupala japansku firmu FUJITSU. Tamo je ucestvovao u uvodenju vise sistema i obrazovanju kadrova. Pored ostalog projektovao je Zajednicki informacioni sistem SR Srbije u socijalno zdravstvenoj zaštiti i Informacioni sistem budžeta Srbije. Sada radi u Zavodu za ekonomske eksperize iz Beograda na poslovima projektovanja AOP sistema.

Centralna memorija

Kapacitati centralne memorija računara instaliranih u radnim organizacijama mere se uglavnom megabajtovima. Računari izgradeni oko osmobitnih mikroprocesora a kojih je daleko najviše omogućavaju programaru korišćenja 64 kilobajta po jednom radnom mestu (terminalu) — isto toliko koliko i mi imamo na stolu kod većine kućnih računara. Šta više sedamdesetih godina, kada je jedan bajt centralna memorija koštao približno 1 dolar samo ja mali broj nas koji amo radili na sistemima za obradu podataka imao sreću da radi sa 64 kilobajta. Da se ne vracamo u kasna šezdesela kada je pojave sistema UNIVAC 1004 sa 1 kilobajt centralna mamorije za nas bila pravi događaj.

Jedinice sa direktnim pristupom

Ukupni kapacitati jedinica sa diraktnim pristupom kod velikih siatama iznose nako-

Iako se u poslednje vreme za kucne računre nalazi i u kući ponoski ozbiljniji posao, ogromna računarska menažerija — koja se, poput bujice, sliva preko severnih granicau naše domove — uglavnom čami neiskorišćena, a novopečeni programeri nisu ni svesni kakvo možno orude imaju u reci. Kućni računari su, naprosto, idealni za malu privredu, a mogu, na odredenim poslovima da nadu svoje mesto čak i u najvećim radnim organizacijama — bez obzira da li u njima već postoji računski sistem ili ne. To, možda, najbolje znaju oni dečaci, za sada latina veoma retki, koji na sovjim skromnim kuchim mašinama obrcu miliono vrsoci obradu podataka za radne organizacije. Ovim tekstom hocemo da odgovorimo na pitanje „zašto da", a za sledeće brojeve „Računara" pripremamo seriju napisa koji će kroz niz praktičnih, konkretnih primera pokušati da odgovore i na pitanje „kako".

liko stotina magabajtova pa i više gigabajtova Kod kućnih računara možamo danas u najboljem slučaju da imamo 10 magabajta na vinčestar disku Vidimo da je odnos veoma napovoljan za kućna računara Ali ne treba zaboraviti da mi na žalimo da koristimo svoj računar da radi sve poslove vac mu namenjujamo samo nake

Magnetna trake

Magnetna traka je varovatno, jedinica koju nikada nećemo imati uz svoj kucni računar Tazlog za ovo je svakako anor mna cena jadinica magnetna trake, ali i činjanica da su danas u velikim računskim centrima magnetne trake degradirane u svojoj funkciji pa služe ugtavnom za čuvanja arhivskih datoteka Ovu funkciju može na kućnom računaru u potpunosti da preuzma na sebe kasetna jadinica Odnos brzina pranosa podataka izmadu disk-jedinice i magnetne trake kod velikih sistema i brzine prenosa podataka izmadu disketna i kasatne jedinice je približno isti Medutim čuvanje podataka kod kućnih računara je mnogo bolje vršiti kopiranjem diskata tako da nam kaselna jadinica nije ni potrabna za poslove koje namaravamao da radimo

Štampač

Svi poslovni računari su opremljani linijskim štampačima koji mogu da štampaju izmedu 300 i 2000 redova u jednoj minuti pri čemu u jednom radu može biti obično 132 znaka Na žalost, o ovakvim brzinama štampanja na kućnim računarima možemo samo da čeznamo Medutim na traba zaboraviti da je skromnih nekoliko desetina znakova u sekundi koliko imamo na našam mališi u potpunosti poradljivo sa brzinama koje imaju malični štampači koji se danas nude na velikim sistemima kao hardcopy uz udaljene terminala

Komunikacione veze

Po pravilu svi savramani računari za obradu podataka su komunikaciono orijentisani To znači da mogu komunicirati bilo izmedu saba bilo sa udaljanim terminalima korišćenjam javnih ili iznajmljenih veza za prenos podataka Medutim danas je lakode moguće formirati i mrežu uz korišćenje kućnih računara Povazivanja kućnog računara sa velikim sistemom ja u potpunosti ostvarljivo Na taj način smo u mogucnosti da za naše potrebe koristimo velika datotake smaštena na jadinicama se direktnim pristupom u okviru velikog sistema tj da kucni računar koristimo kao inteligantni terminal velikog sistema Mada ovej način končenja nija naintaresantan (naprotiv!) — on ipak na predstavlja onu koju žalimo da dodalimo kućnom računaru

Programski jezici

Na velikim sistemima se uglavnom koristi kobol PL/I I tortran a u mnogo manjoj mari Algol, bejzik i paskar Autoru nija poznat ni jedan slučaj da je na velikim sistamima kod nas u zamlji instaliran najnoviji programski jazik ADA koji je nedavno u Sjedinjenim Američkim Državama zbog svojih kvalitata na mnogim mestima prihvaćen kao jedini programski jazik Kućni računari, koji rada pod operativnim sistamom CP/M imaju na raspolaganju sve programske jazika koji su razvijani za velike sistema uključujući i programski jazik ADA Korišćenjem kućnog kompjutera nismo nimalo handikapirani u pogledu raspolozivih programskih jezika

Šlo se tiče pomoćnih programa i kod kucnog kompjutera nam ja na raspolaganju osnovni — SORT/MERGE program Takode nam je na raspolaganju i veliki broj drugih programa za prepis datoteka izmanu imena datoteka testiranje programa i mnogi drugi

Brzine rada

Brzina rada velikoh sistama je nasumnjivo znatno veca od one koju imamo na kucnim računarima Ali za poslove koje želimo da radimo na malim računarima rad u realnom vramanu nije imperativ (uostalom kao i kod velikog broja poslova koji se rada na velikim sistamima) Jedino što želimo je da sa obrada vrši dovoljno brzo — mnogo brža nago ako bismo radili bez računara ako nako od čitalaca ima mogucnost da izvrši standardni banč-mark test za kućna računare na velikom sistemu (autor je to uradio na sistemu NCR 90/20 pri čemu je da bi rezultati bili što verodostojniji u sistemu radio samo tastirani program) varovatno će doci do rezultata koji mogu samo da posluze kao reklama za kucne računare

Tačnost

Pod ovim pojmom svakako na podrazumevamo da li ca i kućni računar sabrati 2 i 2 i kao rezultat dati 4 Sigurno da hoce Ali broj značajnih cifara sa kojima on radi je svakako vaoma važan Za veliki broj primena davat cifara koliko prikazuje na primar amstrad u bejziku je sasvim dovoljno Što se tiča prikazivanja prako mantise i aksponenta 10 na 38 stepen ce takode zadovoljiti veliki broj nasih potreba

Mesto pod suncem

Iz svaga ovoga moguće je dosta kriticki odrediti poslove koje u poslovnoj primani možemo poveriti kucnom računaru Pri tome uopšte ne treba razgovarati o njegovoj primani pri projektovenju Njegovo masto je svakako na radnom stolu inžanjara Ali šta je sa ostalim delovima radna organizacija? Pri tome ne treba zaboraviti da nisu sve radne organizacija veliki sistami sa više hiljada zaposlenih već da ih ima dateko viša sa nekoliko stotina radnika relativno malim brojam osnovnih sredstava nakoliko stotina kupaca i dobavljača Postoje dva osnovna kriterijuma koji određuju da li se neki posao može poslaviti na kućni kompjuter

- veličine datoteka koje će se koristiti
- obim štampe na izlazu nakon završena obrade

Od ova dva kriterijuma drugi zaslužuje posebnu pažnju ali i prvi ima svoju težinu Pogodnom organizacijom podataka na različitim diskatama (na primer kupci su podaljani po azbučnom redosladu) ovaj problam je mbguće uz smanjanja komfora prevazići Ovo će za mnoge moja kolage izgladati kao jaras ali naka se sete samo knjigovodstvenih mašina sa magnatnim konto karticama koje i danas naprikosnoveno vladaju u mnogim radnim organizacijama Rašenje sa kucnim računarom ja ipak mnogo bolje

Medutim problam obima štampe ja nemoguća prevazići Skromna brzina matričnih štampača je u mnogome limitirajuci takior primane kućnih računara u radnim organizacijama Stoga je pra nego što se pristupi izradi pojedinih programskih rašenja najpra potrabno ustanoviti obim štampa koji je potrabno izvršiti u nakoj jedinici vremena dovoljno kratkoj da zadovolji naše potrabe

Što se tiče same konfiguracija kucnog računara za primenu u radnoj organizaciji dilama nama mnogo Pored samog računara nužno je raspolagati bar se jednom jedinicom diskata i jednim matričnim štampačam Druga disketna jadinica će u mnogome olakšati život kako onoma ko buda razvijao samu aplikaciju tako i osobama koje će koristiti iako razvijeni sistem

Cilj ovog članka nija da pokaže kako je postojanje velikih sistema bezpredmatno kada su nam na raspolaganju kućni računari Na protiv danas u našoj zamlji nama ni iz daleka onoliko velikih i modernih sistama koliko bi nam bito potrabno Utoliko pre u svakoj organizaciji kucni računari imaju avoje mesto pod suncem Uostalom krajam sedamdasatih godina firma Taksas Instrumants sa svojim modalom 900 napravila je izvanradan uspah namenjujuci ga upravo onoma o čemu govori i ovaj članak

Ukoliko ovaj tekst navada bar jadnog čitaoca da razmišlja o mogućnostima primene kucnih računara u radnoj organizaciji i ako kao posladica toga, bar jedan računar zaživi nagde autor će smatrati da je njegov trud u potpunosti nagraden, a da je članak ispunio svoju namenu

Mihailo Karapandžic

Spektrum Programator eproma

Pre nekoliko godina u Radio-amaterskom magazinu 2 objavljen je projekat programatora eproma za spektrum i ga leksiju sa odgovarajućim softverom za prvi i najavom softvera za drugi računar. Čitaoci koji su pokušali da unesu onih nekoliko stotina bajtova mašinca nash su se bez sumnje u velikom cudu — objavljeni program je predstavljao prosti zbir besmislenih bajtova kao da je pre ulaska u štampu propušten kroz mašinu za šifriranje. Neši mladi tehnički saradnici, koji su listali ovaj program na štampaču uzimali su malo od izvornog a malo od objektnog koda dok program nije izbližen do neupotrebljivosti.

Detaljna uputstva za pisanje programa koja smo dali u računarima 3 malo su ublažila čitavu stvar ali je prema graditeljima programatora eproma ipak ostao veliki dug koji isunjavamo ovom prilkom. Pored integrisnog programiranja tipa čitav eprom odjednom nova verzija programa omogućuje i parcijalno programiranje tipa bajt po bajt. Ova verzija radi besprekorno više od devet meseci — sa njom je isprogramirano oko 2000 eproma tipa 2732 s najmanje po stotinu od svih ostalih tipova. Nijedan amaterski projekat se izuzetkom razume se računara galaksija nije podvrgnut tako rigoroznim testovima kao programator eproma.

Iako u komercijalnom softveru, nažalost često kod računara najnovije generacije polako preovladuje princip pokaži prstom ono što hoćeš prilikom razvoje programa uložen je poseban trud upravo da se izbegne sistem menija. Sa dosta razloga. Šanse da se izborom opcije iz menija unese pogrešna naredba ili pogrešan podatak znatno su veće nego u komandnom režimu kada se oni ne biraju nego se unose svojeručno. Izvesten broj grešaka u radu sa programatorom eproma može zaista da bude štetan po zdravlje.

Zato je spektrumov operativni sistem proširen specijalnim setom naredbi za rad sa programatorom eproma. Naredba se pozneju po zvezdici ispred imena unose se slovo po slovo i mogu ali i ne moraju da imaju odredene parametre. Ako se parametri izostave program ih dodeljuje sam —

EPROMER

Najpre otkucajte hex-loader 1 startujte ga s enda, red po red, unesite masinski program. ENTER pritiskatā tek na kraju svakoga reda tj. posle otkucate devet bajtova. Osam od njih će biti upisani u memoriju dok je deveti zaduzen za proveru korektnosti kucanja.

CTRL coda hex listi

```
25500: C3DA633A3A5CFE00D9
25508: 2034FDCB01/E204ACE
25516: 2A025C233E5C3B/12CC
25524: CD0313FD3600FF2A3F
25532: 595CCDA7112A025C72
25540: 23E5C3B412FD3600C4
25548: 002A5D5C225F5C18E5
25556: D1225D5C19EF2A5D3A
25564: 5C207EFE2A20C03223
25572: 5D5CFD3600FFFD361E
25580: 2600067110A602A90
25588: 5D5C23131A3C2807D
25596: 3DBE20F61A133C20A2
25604: FB3D28CD051B1BE74C
25612: 7803204B225D5CFEE2
25620: 022854/EFE0D280433
25628: FE3A200CFD3600FF96
25636: FDCB017E2805C1833BE
25644: CD7A1CF53A0450FEEF
25652: 033043F1FE2CC2C926
25660: 43E7CD0211CFDCB007D
25668: 7ECAC963FE00D200900
25676: FD3600FE3CC2C901
25684: 43FD3600FFFDCB015E
25692: 7E202A2A025D23E500
25700: 210712E5C3761BDFE0
25708: 021CFE4128002107380
25716: 50C6BEC3416AF11855
25724: C4210750CDFEE7C3BA
25732: 4160EFA03821075BEF
25740: CB06BAB450FE01CAE3
25748: FE60FE0228ECFE03F7
25756: CAFE64F53E02CD012F
25764: 163E011118D69CD0A8E
25772: 0CF13D113F6ACD0ACB
25780: 0CCD5911E70B1CAC4E7
25788: 65C53E0E321050E51
25796: 04CDB365E1220A5B51
25804: CD991EC53E0C3210D5
25812: 5B3E03CDB365E1220
25820: 02502280E5B3A045B81
25828: FE04CA0666CD991EC2
25836: C53E0A321050B3E02EA
25844: CDB365E1220456C306
25852: 0C662A0E25C23E5C375
25860: 7610CD991EC53E0A22
25868: 602806316989C2D5DC
25876: 63C5E106057DD210F
25884: E169DD0E002009DDF1
25892: 23052BF6C5D5637001
25900: FE9420802107750CB70
25908: 7E20013D32065BC13E
25916: F53E0632105AFCD52
25924: 0365210756C07EC4AB
25932: 7DA8F13DC9C621E6AB
25940: 4816005F195F21EB60
25948: 6819220805BC5FE6020
25956: 3A0458FE04281221F6
25964: 307522005BE5C13E06
25972: 0A32105B3E02CD0367
25980: 6521E0002202502227
25988: 0E5BE5C13E0C32109B
25996: 503F3CDB365DD2A09
26004: 0658DD6E07DD660060
26012: 220A5E5C15E0E32A0
26020: 105E3E04CD636521B3
26028: 0750CDFE4C30C66C51D
26036: C5F553E02CD0116F1CF
```

```
10 REM CTRL CODE HEX LOADER

20 BORDER 1: PAPER 1: INK 7: P
OKE 23609,20

30 CLEAR 25499: POKE 23658,0:
PRINT AT 20,0

40 INPUT "START ADDR:";A

50 INPUT (A);": "; LINE A$: IF
A$= " THEN STOP

60 LET C=0: IF LEN A$<>18 THEN
GO TO 90

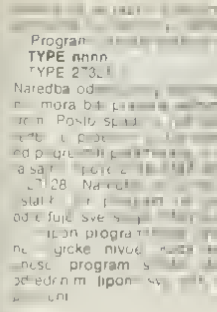
70 FOR I=1 TO 17 STEP 2: LET X
=16*CODE A$(I)-748-112*(A$(I)>"9
")+CODE A$(I+1)-48-7*(A$(I+1)>"9
"): IF X<17 THEN POKE A,X: LET A
=A+1: LET C=C+X

80 NEXT I: LET A=A-8: IF C-256
*INT (C/256)=X THEN PRINT A;": ";
A$: LET A=A+8: GO TO 50

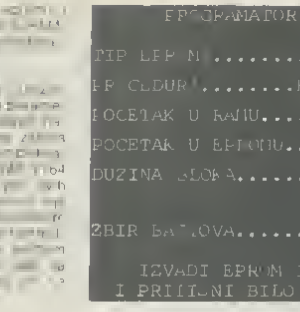
90 PRINT 0;: FLASH 1;"ERROR!";:
BEEP 1,1: GO TO 50
```

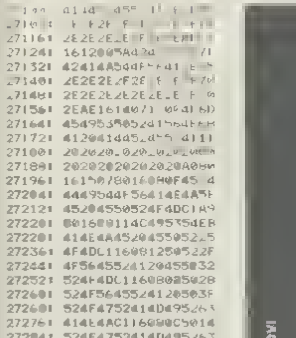
```
26044: 111969CD0A8CC11E54
26052: 8521F0D0883E161D62
26060: 2118FC09380F1D21C3
26068: 9CFF09380E1D21F618
26076: FF0930011D3E20934F
26084: 210756CB7E2012F5F3
26092: 3E16D/3A1050D7F198
26100: D7C1CD2920DCDE329A
26108: C9F53A0458FE0220/F
26116: 03F11B3E3F13D10DF14
26124: 3A045BFE07CA1A67E9
26132: FE0528ECCD9967CD31
26140: A567CDAF67CD6167A4
26148: CDBA67DD7E01D33F5C
26156: DB1FFEFFC2C966CD87
26164: 6567CD7067A0F1DD5E
26172: 7E00D33F3A0450FE27
26180: 04CADA66CD9967CDBA
26188: AA67CDAF67CDBA67E2
26196: DD7E02D33F7EFEFFEA
26204: CA5F67D31FDD7E03E0
26212: D33FCD1269F3C53F4F
26220: 150ED80D20FD3D2085
26228: F0C1FBDD7E02D33F25
26236: CD6567CD7067D0532
26244: ED50025BFD530E5E4F
26252: 2107500E0AECD9967C9
26260: CDA567CDAF673A04FA
26268: 5BFE0520035CD81673
26276: CDBA67DD7E01D33F5C
26284: D01F0E2838CD6567A9
26292: CD7067CD7567D20F0D
26300: 11BE6ECDD567CDF050
26308: 67CDFA67C3FE6011C0
26316: DC6ACDD567CDF06773
26324: CDFA67C3FE6411B91D
26332: 6ACDD567CDF067CD6A
26340: FA67C3FE64F5210793
26348: 50C06E2010E11126E9
26356: 60CDD567CD8F6BCD85
26364: 8160CD3069CDE16763
26372: FE0D2809CDFA67C32D
26380: FE64E118EAE5210752
26388: 5BCBEEE11097CD990A
26396: 67CDA567CDAF67CDF0
26404: 8167CDBA67DD7E0132
26412: D33FAF320C5BDB1F54
26420: 773A0C5B863208C5837
26428: CD6567CD7067E20E40
26436: DD7E00D33F3E1232EE
26444: 1050E0803A0C50d761
26452: 3E06CD0365CDFA6757
26460: C3FE64CD7567C37C0D
26468: 46DD7E06D35DDD7E52
26476: 05D35DC923807B0155
26484: C9C5ED5BD0F5003ED1C
26492: 43E05B01C9DD7E0091
26500: D33F11F46ACDD56790A
26508: 116460CDD567CDE197
26516: 67CD6166C92A0050C6
26524: ED4B0A50DD2A0050D7
26532: C93E90D37FC93F9070
26540: D37FC9DD7E04D35DAA
26548: DD7E05D35DC9F5215F
26556: 075BC876CCC567E17C
26564: C9EDSB02587AB3C863
```

```
26572: CD65671B7AB320F8F9
26580: C9AF323C5C1AFE0865
26588: C8D71318F976FDC000
26596: 816E28F9FDCB01AE07
26604: 3A095CC9CD8F6001/C
26612: 9600CD3D1FC9DD7E03
26620: 00D33F114660CDB576
26628: 67116606CDD567CD1D
26636: E167CD0168C9E5C971
26644: DDE53E10321050FE0
26652: 05ED4B0E5BC5CDD3E9
26660: 65C103EDB430E50DD9F
26668: E1C1E1C9E5C5DDE5B0
26676: E5E53E07111869CDBE
26684: 0A0CC1CDF665E17E5E
26692: CD5060E2E0D7ED4BFD
26700: 0E5BCDF665DB1FCD5B
26708: 5068DDE1C1E1C9F5E1
26716: 3E2ED7F1F5F6F01F10
26724: 1F1F1FC630FE3A38C3
26732: 02C687D7F1E607C652
26740: 30FE3A300AC607D746
26748: C93Ea1D7C9E5C5DDEF
26756: E5064CD4404EDDE1CC
26764: C1E1C9E5D5DDE506ED
26772: 0AC521000011640065
26780: E5CD0503011400110
26788: 640AE1C600ED4AE527
26796: C600ED52E130E6C1C5
26804: 10FDDDE1C1F1C946A4
26812: 4F4144FF5341564502
26820: FF564552a94659FFD3
26828: 54d55354FF4C495327
26836: 54FF545950455FF504
26844: 4F4B45FF00CACACD2
26852: CCF8241612050041SF
26860: 810746D4865000864
26868: 50905E1E4040504FF9
26876: 1050905C1C0400405F5
26884: 80107B0E41484046d
26892: 0500207000341420F
26900: 02030400816030E4
26908: 50524F4752414D4159
26916: 544F5220204550521C
26924: 4F4D410D0D0D5449A1
26932: 5020455052AF4D4134
26940: 2E2E2E2E2E2E2E70
26948: 2E2E2E2E2E2E2E70
26956: 2EAL1600405024EE
26964: 4345445552412E2E19
26972: 2E2E2E2E2E2E2E70
26980: 2E2E2E2E2E2EAEF0
26988: 160A0450F4345554D0
26996: 4146205520524146D1
27004: 552E2F2F2E2L2E2F97
27012: 2E2E2E2E2E2F2E70
27020: 2EAE160C00504F43E0
27028: 4554414B20552045FF
27036: 50524F4D5522E2E1D
27044: 2E2E2F2E2F2F2F2F78
27052: 2E2E2EAE160E004A0
27060: 555A494L412042AC35
27068: 4F4B412E2E2E2E2EC1
27076: 2E2E2E2E2E2F2F2E70
27084: 2E2E2L2E2EAE1610BA
27092: 00544554E5554142DD
```



Program
TYPE nnnn
TYPE 2732
Naredba od
mora bi
Posle st
od pr
a sa
128. Na
sta
od
ipon progra
gr
pro
od
ni





33) programatat eproma

- **LOAD/*LOAD xxxx nnnn mmmm**
  LOAD *LOAD 45000 200 1000
  Kopira master eprom u RAM integralno (od adrese 0 u epromu na adresu 30000 u RAM-u) ili samo jeden blok (od adrese nnnn u epromu na adresu xxxx u RAM u) čije je dužina određena parametrom mmmm. Uneta bez parametara naredba LOAD ima isto dejstvo kao i da je unelo LOAD 30000 0 4096. Preslikavanje mastera u RAM se završava porukom o zbiru bajtova u masteru. Zbir se izračunava u toku kopiranja po modulu 256. Kopirenje mastera u RAM razume se ne predstavlja uslov da bi se obavilo programiranje — u eprom se može preneti sadržaj sa bilo koje lokacije u memoriji računara između 0 i 65535
- **SAVE/*SAVE xxxx, nnnn, mmmm**
  SAVE SAVE 45000 200 1000
  Upisuje sadržaj RAM-a u eprom integralno (sa adrese 30000 u RAM-u na adresu 0 u epromu) ili samo jedan blok (sa adrese u RAM u na adresu nnnn u epromu) čije je dužine određene parametrom mmmm. Uneta bez parametara naredba SAVE ima isto dejstvo kao i da je unelo *SAVE 30000 0 4096. Pre ulaska u proceduru programiranja program vrši automatsku proveru da li je eprom prazan a po završenom programiranju automatsku verifikaciju. Direktno programiranje bez ovih verifikacija programski je potpuno zebranjeno. **Od programiranje ćelije u kojoj** se već nalazi neki sadržaj moze de bude samo stele. Programiranje je linearno — svaka lokacija se programira 50 ms. Preskaču se samo one lokacije u koje treba upisati FF jer je to početni sadržaj svake prazne ćelije u epromu. Operacija programiranja se završeva zvučnim upozorenjem i porukom

PROGRAMIRANJE USPELO
ili
PROGRAMIRANJE NIJE USPELO

- **TEST/*TEST nnnn mmmm**
- **TEST/*TEST 1000 2000**

Proverava da li je prazan ćilav eprom ili samo jedan određeni segment (od adrese nnnn do adrese nnnn + mmmm). Ova opcija je posebno pogodna za one koji često brišu svoje eprome. **Operecija se, završeva porukom**

EPROM ISPRAVAN
ili
EPROM NIJE PRAZAN

- **VERIFY/*VERIFY xxxx, nnnn, mmmm**
  VERIFY VERIFY 45000 200
  Upoređuje sadržaj u memoriji računara (master) sa sadržajem u epromu koji se trenutno nalazi u programatoru. Ako u epromu pronađe lokaciju čiji se sadržaj ne slaze sa sadržajem odgovarajuće lokacije u RAM-u, **program ispisuje** heksadecimalno obe lokacije njihove sadržaje i zaustavlja verifikaciju. Nastavak se obezbeđuje tasterom ENTER a potpun prekid i izlazak iz procedure provere pritiskom na bilo koji drugi taster.

Nove naredbe su gotovo savršeno uklopljene u operativni sistem i bejzik interpreter računara — spektrum. Ih u po čemu ne razlikuje od svojih sopstvenih. Naredbe se mogu izdavati direktno se tastature ili iz programa. Dozvoljeno je slobodno mešanje or ginalnih i novih naredbi u jednoj jedinoj komandnoj liniji i velikom bejzik programu. To praktično znači da se programator eproma može pomoći i svim potencijalima spektrumovog bejzika. Prilikom programiranja ROM ova za računar galeksije obično je korisćen program poput ovog

```
10 RONDOMIZE USR 25713
20 TYPE 2732
30 LOAD
40 FOR A  1 TO 20
50 SAVE
60 NEXT A
```

Ovaj mali program kopira master eprom u memoriju računara počev od adrese 30000 i programire dvadeset kopija. Programer treba samo da menja eprome.

Za uvođenje novih naredbi iskorišćena je tehnika preseretenja rutine za obradu grešaka na čiju adresu pokazuje promenljivi ERR-SP. Nekon inicijalizacije programa na adresu na koju pokazuje ova promenljiva ubacuje se njegova sopstvene adresa. Pre nego što se vrati u bejzik interpreter računara program svaki put obnavlje ovu adresu i zahvaljujući tome ostaje stalno aktiven lačnije sve dok se ne pređe na upravljanje iz programa u bejziku. Pre pozive prve nove naredbe u programu mora obavezno da se nađe jedno RANDOMIZE USR 25713 bez obzira što je program jednom već inicijalizovan. To je relativno skromna cena za jednu veoma efektnu tehniku uvođenja novih naredbi.

Rad bez menija ima međutim i **svojih nedostetaka — programer nema** uvek potpuni uvid u tok programa. Sloga je uveden statusni ekran (vidi sliku) iz koga programer u svakom trenutku može da vidi sa kojim tipom eproma radi koji je master u računaru koliki je njegov zbir bajtova koja je operacija u toku pa čak i koja se adresa trenutno programira. Nekon unošenja svih podataka program zahteva od korisnika da poslevi eprom u podnožje i kao znak da je to učinio pritisne bilo koji taster — poslednja **prilika da se predomisli i ispravi eventualne greške.**

Skladnim uklapanjem u interpreter bejzika i opcijama za parcijalno programiranje ovaj program pretvara programator eproma iz sprave za kopiranje (tudeg truda) u spravu za razvoj sopstvenog sistemskog softvera.

Jova Regasek



„Spektrum"

SPRAJTOVI

Sigurno vam je već dosadilo da svaki „komodorovac" kojeg sretnete na ulici sruši ugled vašem miljeniku onom poznatom Ali komodor ima sprajtove! No na očajavajte jer ovaj program ispravlja taj „spektrumov nedostatak Ukucajte ga otipkajte RUN I kad „spektrum kaže OK, sprajtovi čekaju na vas Program možete spremiti sa SAVE – sprajtovi CODE 64032 264

Program omogućava korištenja osam sprajtova, obilježanih brojevima 0–7 i veličina 16×16 točkica Najveći prioritat ima sprajt 0 a najmanji sprajt broj 7 — pri preklapanju sprajtova sprajt obilježan malim brojem bit će nacrtan preko sprajta obilježenog većim

Upravljanja svakim sprajtom postiže se kao i kod „komodora pokretanjem na četiri posebne memorijske lokacije Za sprajt 0 to su lokacije 64000 64001 64002 i 64003 za sprajt 1 lokacija od 64004 do 64007 itd do sprajta 7, čija su lokacija od 64028 do 64031 Prva od te četiri lokacija indicira je li sprajt uključen (u tom slučaju mora sadržavati broj 1) ili isključen (kada sadrži broj 0) Druga lokacija sadrži X, a treće Y koordinatu gornjeg lijevog kuta sprajta Četvrta lokacija sadrži atribut boja za sprajt

Da bista definirali sprajt, prvo ga nacrtajte na milimetarskom papiru u veličini 16 sa 16 mm Zatim ga podijelite u dvije kolone široke 8 mm Prvu kolonu ukucajte u grafičke karaktera A i B– s drugu u C i D Na lokaciju 64316 unesite broj sprajta koji definirate i otipkajte RANDOMIZE USR 64240 Sad je taj sprajt spreman za upotrebu a sadržaj grafičkih karaktera više nije važan

Za iscrtavenja svih uključenih sprajtova dovoljno je otipkati RANDOMIZE USR 64032 No ne zaboravite za svaki sprajt unijeti odgovarajuće podatka u njegova 4 lokacije

Evo jednog malog ali korisnog sevjete uz rub sprajta ostavite bar 1 red točkica razmaka kako sprajt ne bi prilikom pomicanja ostavljao trag

S obzirom da upravljanja sprajtovima pomoću naredba POKE nija baš elagantno evo zadatka za prava hakera obogatite spektrumov bajzik naredbom za upravljanje sprajtovima

Igor Fišer

```
  10 FOR i=64032 TO 64315
  20 READ a: POKE i,a
  30 NEXT i
1000 DATA 62,7,50 60 251,33,250,
7
1010 DATA 7 111,125,61,250,232,2
50,35
1020 DATA 78,35,70 120 254,176,2
10,232
1030 DATA 250,63 31 55,31,183,31
168
1040 DATA 230 246 168,87,121 254
,240 210
1050 DATA 232 250,197,7,7,7,168,
230
1060 DATA 199,168,7,7,96,35,126,
6
1070 DATA 121 230,7,245 38,0,15,
15
1080 DATA 15,111,15,79,68,9,229,
56
1090 DATA 60,251,103,48,0,77,31,
203
1100 DATA 25,71,9,1,0,230,9,193
1110 DATA 9,193,4 62,255,163,31
16
1120 DATA 252,23,79,6,16 213,121
,47
1130 DATA 79,26,161,192 18,35 20
,126
1140 DATA 16,35,28,121,47,79,26,
161
1150 DATA 192,16,35,209,20,122,2
30,7
1160 DATA 194,175,250,123,196,32
,95 230
1170 DATA 224,40,4 122,214,6,87,
16
1180 DATA 212,193,121,230,7,95,1
21,7
1190 DATA 7,7,168,230,199,168,7,
7
1200 DATA 111,120,7,7,230,3,246,
80
1210 DATA 103,6,87,14,3,120,230,
7
1220 DATA 194,212,250,13,65,114,
30,114
1230 DATA 36,175,170,40,1,114,12
5 196
1240 DATA 30,111,124,205,0,103,1
8,237
1250 DATA 58,60 251,61 242,34,25
0 201
1260 DATA 33,120 255,6 16,54,0,3
5
1270 DATA 16,251,50,60,251,230,7
,40
1280 DATA 0,93,103,31,203,27,87,
28
1290 DATA 17,0,230,25,221,33,88,
255
1300 DATA 8,16,221,126,0,119,35,
221
1310 DATA 126,16,119,35 221,126,
32,118
1320 DATA 36,221,35,16,237,64,93
,1
1330 DATA 46,0,237,46,1,60,1,6
1340 DATA 126,31,16,35,19,6,11,1
20
1350 DATA 177,32,244,201
```

„Spektrum"

SCREEN (Y,X)

„Spaktrumova funkcija SCREEN(Y X) ima jedan prilično ružan bag na radi za UDG i blok grafičke karaktera Program bata bajzik (VI 8) ratina otklanja taj bag ali ni to rašenja nije savršano blok grafički karakteri i dalja bivaju ignorisani Stvarno potpuno rašenja predstavlja ovaj kratak mašinski program smešten u baferu za štampač Program se izuzatno lako koristi U bajzik varieble X i Y stava se koordinate polja na ekranu koja treba lapitati Zatim USR 23300 daje kod znaka koji se tu nalazi Jedan flag na adresi 23540 ukazuje na to da li je prepoznati znak u normalnom (0) ili inversa (1) obliku Prilikom upotrebe ovog programa traba obratiti pažnju na nekoliko sitnica. Znak koji nija prapoznat dobiće kôd 0 Zatim, grafički karakter sa kodom 128 (prazan kvadratić) neće biti prapoznat kao takav već kao razmak (32), što je uostalom i sasvim logično Takoda grafički znak 143 (puni kvadratić) biće prapoznat kao inverzni razmak, dakle kôd 32 i flag postavljan na jedinicu. Ovaj program ima još jednu caku moguće je ispitivati i donji dao ekrana Naime, Y koordinata može da ide od 0 do 23

Ako je ovu rutinu potrebno pozvati iz nekog mašinskog programa, onda je ulazna tačka na 23325 U registru B treba da se nalazi horizontalna koordinata (0 do 31) a u registru C vertikalna koordinata (od 0 do 23) Na izlazu, kôd znaka se nalazi u registru C Flag se naravno i dalje nalazi ne 23540

Kada otkucate program snimite ga na kasetu sa SAVE „SCREEN CODE 23300 245 Prilikom učitavanja, nija potrebno nikakvo spuštanje RAMTOP-a Treba samo paziti da se tokom upotrebe programa na pozovu slučajno naredbe COPY LPRINT ili LLIST koje brišu bafer za štampač

Vladimir Kostic

```
SCREEN$(Y,X)
CTRL CODE HEX LIST

23300  21F55BCDD6SBFE20BD
23308  3802CF01F521F75B72
23316  CDD65BFE1830F3C1FB
23324  4F2A365C1100011936
23332  3E60CD8B5B385C2H0F
23340  7B5C3E15CDB85B300D
2334B  04C625184E11001076
23356  3E1832AA5BAFCB4B52
23364  28023EF0CB432B0290
23372  C60F676F22EH5B2234
23380  EC5BAFC05B2B023EB4
2338B  F0CB532802C60F6774
23396  6F22EE5B22F058216B
23404  EA5B3E01CD885B3067
23412  057BC6801806C1C1515
23420  20C33E00F53E2032H6
2342B  AA5BF106004FC9C5D9
23436  D5F5AF32F458790F82
23444  0F0FE6E0AB5F79E64A
23452  18EE4057F147L5D56F
23460  E51AAE28083C201E5H
23468  3DF53E0132F45BF1E3
23476  4F0607142314AAEA904
23484  200C10F7C1C1C13EB4
23492  809037D1C1C9E11194
23500  080019D1C110CFAF41
2350B  18F1ED5B5D5CD52201
23516  5D5CCDFB24D1ED53B6
23524  5D5CCD1423C90000B6
23532  0000000000000000
23540  00580D590D000000C8
```

3. STACK_A — CALL &2D28

Potprogram STACK_A će preneti na računski stek broj sadržan u akumulatoru. Izlaznih parametara nema a reg.str. nisu očuvani.

4. STACK_BC — CALL &2D2B

Ova rutina za razliku od pretnodne prenosi na računski stek broj sadržan u BC. Inače i STACK_A sama prvo izvrši **LD C,A** i **LD B,&00** pa dolazi na STACK_BC. Izlaz je tako zajednički — registri nisu očuvani.

5. STACK_NUM — CALL &33B4

Potprogram STACK_NUM donosi na računski stek proizvoljan broj iz memorije prikazan u adresiranom zapisu sa pet bajtova. Na ulazu HL sadrži adresu prvog bajta.

U toku potprograma akumulator je očuvan. BC na izlazu uvek sadrži nulu a HL će sa pomeriti iza broja koji je bio prenet. DE sadrži adresu STKEND.

Najčešće se STACK_NUM koristi kada se na stek šalju elementi nekog numeričkog niza. Čim se nađe adresa zadatog elementa(pomoću STK_VAR) izvrši se **INC HL** i odmah zatim STACK_NUM. Na izlazu će HL pokazivati sledeći element niza spremen za novo STACK_NUM.

6. STK_VAR — CALL &2996

O ovoj rutini smo već govorili prilikom traženja adrese elemenata numeričkog niza. Međutim, isti potprogram se koristi i kada na računski stek treba poslati parametre nekog alfanumeričkog niza.

Na ulazu već smo to rekli, STK_VAR koristi izlazne parametre iz LOOK_VARS. Ako je pri tome promenljiva bila znakovnog tipa, njeni elementi će se ubaciti na stek. Nikakvih izlaznih parametara neće biti i registri ne ostaju očuvani.

7. STK_STORE — CALL &2AB6
STK_ST_0 — CALL &2AB1

Potprogram STK_STORE šalje na računski stek pet bajtova smeštenih redom u registre A, E, D, C i B. Na izlazu su registri očuvani a HL sadrži adresu STKEND.

Najčešće se na ovaj način prenosa parametri niza. DE treba da sadrži adresu a BC dužinu niza. Možete sa pozvati i SKT_ST_0 koji će automatski staviti nulu u akumulator. Name taj prvi bajt ionako nije važan.

FP_TO_A — CALL &2DD5
FIND_INT1 — CALL &1E94

Poslednji broj sa računskog steka može se preneti u registar BC pozivom rut ne FP TO_BC. Ulaznih parametara nema. Na izlazu BC sedrži traženi broj a njegov predznak je odreden indikatorom nule ,Z ako je Z=0 broj je negativan. Poslednje vrednost sa računskog steke bive automatski obrisana. HL pokazuje ispred nove poslednje vrednosti a DE iza nje (STKEND). Akumulator sadrži isto sto i registar C.

Ukoliko broj ne može de sa smesti u 16 bita na izlazu ce biti setovan indikator prenosa C. Brojevi koji nisu celi se zaokružuju.

Pozivom FIND_INT2 postići ćemo to de sa u slučaju negativnih brojeva ili brojeva vecih od 65535 automatskt poziva prekid B Integer out of range.

8. FP_TO_A — CALL&2DD5
FIND_INT1 — CALL&1E94

Potprogram FP_TO_A uzima sa računskog steka poslednji rezultat i smešta ga u akumulator. Ulaznih parametara nema a izlaz je potpuno isti kao da smo pozvali FP_TO_BC sa jedinom razlikom sto je indikator prenosa c setovan u slučaju da broj premasuje 8 bita.

Ako se pozova FIND_INT1 onda ce negativni brojevi i brojevi veci od 255 izazvati prekid B Integer out of range.

10. STK_FETCH — CALL &2BF1

Rutina STK_FETCH uzima sa računskog steka poslednjih pet bajtova i smešta ih redom u registre A, E, D, C i B. Ulaznih parametara nema a na izlazu je se steka obrisana poslednja vrednost i HL sadrži novu adresu STKEND.

Obicno se ne ovaj način uzimaju parametri n za adresa DE i dužina BC.

struktura memorije i potprogrami iz ROM-a

sve ‚spektrumove' rutine

Jovan Skuljan

Verovatno nemu programera koji se pre ili kasnije ne zasiti bejzika. Ojade konačno mamutski listinzi i programi koji se iznad svega izvršavaju katastrofalno sporo. Naravno, postoje i situacije kada je bejzik sasvim prikladan i kada od njega ne vredi bežati bez razloga ali to još uvek ne znači da ne treba potražiti bolja i elegentnija rešenja.

U ovom tekstu pokušaćemo da detaljnije upoznamo računar spektrum, odgovarajući na pitanja kako on radi i šta se s njim može postići ako se napusti bejzik.

Pošli smo od pretpostavke da naši čitaoci već imaju kakvu takvu predstavu o mašinskom jeziku tako da im praćenje ovog teksta neće biti naročito teško. Ako i bude nekih problema ne treba se zbog toga obeshrabriti već u sledećem broju. Računari donose svoju školu mašinskog programiranja.

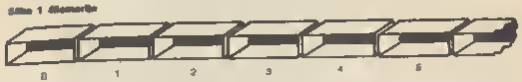
Struktura memorije

Memoriju računara najjednostavnije je zamisliti kao dugačak niz pregradaka ili ćelija koje su u stanju da upamte ono što se u njih stavi. Jedna takve ćelije zove se **bajt**. Veličine memorije kojom računar raspolaže potpuno je određena brojem bajtova.

Jedan kilobajt (1K) sadrži 1024 bajta, a jeden megabajt (1M) 1024 kilobajta. Malo je neobično što se umesto okruglog broja 1000 koristi 1024, ali računar uopšte i ne radi u dekadnom brojnom sistemu. Dodatalom ako se prikaže heksadekadno ili binarno broj 1024 je sasvim okrugao.

Svaki bajt ima neku oznaku pomoću koje se može lako proneći među drugima. Ta oznaka zove se **adresa**, a u stvari je ceo broj u rasponu od 0 do 65535. Prvi bajt memorije ima adresu nula, drugi ima adresu jedan, itd.

Slika 1. Memorija



Češće ćemo adrese i druge konstante označavati heksadekadno, i u tom slučaju ispred njih će stajati oznaka &. (CHR$ 38).

Prema tome, adrese može biti bilo koji broj između &0000 i &FFFF.

Svaki bajt za sebe sastoji se iz osam elementarnih ćelija, koje zovemo **bitovima**. Jedan bit može da upamti samo jednu od dva osnovne informacije: ili je bit **resetovan** (sadrži nulu) ili je **setovan** (sadrži jedinicu). Tade sa čitav bajt može shvatiti kao binarni broj sa osam cifara.

Bitovi u okviru jednog bajta označavaju se indeksima između 0 i 7, onako kako raste odgovarajuća težina cifarskog mesta. Bit 0 je bit najmanje težine.

Slika 2. Bajt



Mikroprocesor može da pronađe bilo koji bajt memorije i da pročita bilo koji bit u njemu. To je čak izvodljivo i iz bejzika korišćenjem funkcije **PEEK**. Baš kao sto **SQR** n deje koren broja n, teko **PEEK e** daje sadržaj bajta ne adresi n. Ne primer **PRINT PEEK 0** će dati rezultat 243 što znači da prvi bajt memorije sadrži vrednost 243, odnosno BIN 11110011, ili heksadekadno &F3. Nijedan bajt ne može sadržati više od BIN 11111111, odnosno 255 dekadno.

Memorija je podeljena u dve velika bloke. Prvih 16K počev od adresa 0, pa zaključno se 16383 čini tzv. ROM (Read Only Memory). To je memorije iz koje se može samo čitati ali u koju se ništa ne može upisivati. U ROM-u su sadržane sve instrukcije potrebne za uspešan rad računara i ta informacija naravno ostaje očuvana čak i kada se računar isključi.

Ostatak memorije, počev od adrese 16384 (heksadekadno &4000) naziva se RAM (Random Access Memory). Prstup tim bajtovima je potpuno slobodan. U njih u bilo kom trenutku možemo upisati bilo koji sadržaj, ali kade se računar isključi, svi podeci iz RAM-a biveju nepovratno izgubljeni.

Naredba **POKE e, b** iz bejzika upisuje vrednost ,b, u bajt na adresi n. Međutim, odmah treba reći da ovakvi eksperimenti nisu preporučljivi, ukoliko nemamo jasnu predstavu o tome šta zapravo radimo. Jer nije nikakav problem ubaciti neki broj u neki bajt, ali treba znati ba i spektrum koristi RAM za sopstvene potrebe. Ako je on na primer na adresu 23607 upisao broj 60, ne možemo sada i mi tamo upisati bilo šta, jer vrlo lako može doći do neporazume.

Potrebno je dakle poznavati strukturu čitava memorije, odnosno tzv. **memorijske mape**. Tada će nam biti jasno šta znači broj 60 na adresi 23607, ili broj 255 na adresi 13610, ili bilo koji drugi bajt u RAM-u i ROM-u.

BROJEVI U POKRETNOM ZAREZU

Jedan bajt dovoljan je za prikazivanje celih brojeva između 0 i 255. Dva bajta, odnosno šesnaest bita, mogu služiti za smeštanje celih brojeva sve do 65535. Ne primer broj 3000 bi

Proizvoljna funkcija f(x) gde je x po modulu **ije veće od jedinice**, može se razviti u red Čebiševljevih polinoma

$f(x) = c_0 T_0 + 2c_1 T_1 + 2c_2 T_2 + 2c_3 T_3 + \ldots \quad |x| \leq 1$

gde su c_0, c_1, c_2 itd. koeficijenti razvoja unapred poznati za svaku funkciju.

Tačnost izračunavanja zavisi od broja članova reda. Spektrum, na primer, računa funkciju SIN sa šest funkcija EXP sa osam, a LN sa čak dvanaest članova. Pri tome je uvek argument podesnom smenom doveden u interval između −1 i +1.

Ako se aproksimacija funkcije vrši redom od ,n, članova, onda se operacije ,sense poziva kodom &80+n. Recimo za red od 8 članova, treba izvršiti naredbu &88.

Iza samog koda operacije mora se navesti svih n koeficijenata počevši od c_0, pa zaključno sa c_n. Za svaki koeficijent se prvo navodi eksponent a zatim mantisa, na potpuno isti način kao u slučaju naredbe ,stk-deta. Podrazumeva se naravno da je pre poziva generatora reda argument , x smešten na stek. Na povratku će on biti zamenjen sa f(x).

Ugrađeni Spektrumov kalkulator raspolaže sa šest **memorijskih lokacija**, označenih sa ,mem-0, mem-1, itd. do mem-5, u kojima se privremeno mogu čuvati međurezultati u toku računanja.

Naredba tipa ,st-mem, će preneti u odabranu memorijsku lokaciju poslednji broj sa računskog steka, mada se sam stak pri tome ne briše. Slično tome, podatak iz bilo koje memorije može se dovesti na stek operacijom tipa ,get-mem. Kodovi za ,st-mem, su &C0+n a za ,get-mem, su &E0+n, gde je n oznaka memorijske lokacije između 0 i 5.

U tabeli računskih operacija namerno smo izpustili neke kodova kao što su recimo &09 — &0E ili &11 — &16. Postoje naredba koje odgovaraju i tim kodovima, ali se ne želost ne mogu koristiti na način koji smo opisali. Očigledno se radi o previdu koji je ato prošao neopaženo kroz sve testove i sade ukrašava ROM vašeg računara. Sva te dodatne operacije, naime, očekuju da u procesorskom B registru zateknu svoj kôd! Pošto se to normalno ne dešava, moramo semi da prvo prekinemo kalkulator (operacije ,end-calc,) zatim de izvršimo **LD B, ee** i onda ponovo startujemo kalkulator, ba bismo izvršili operaciju nn. Baz tog manevra operacija se neće izvršiti korakitno.

Na sreću, ove ekscentrične naredbe nam uglavnom nisu potrebne jer obavljaju obično poređenje dva broja ili dve niza, što se uvek može svesti na ,less-0, i ,greater-0,. Ali, tu speda i jedna dosta važna operacija, vel. Mozde se pitate kako onda spektrum u bajziku uspešno računa funkciju VAL? Ne to nije teško odgovoriti: spektrum pri izračunavanju bilo kakvih izraza koristi isključivo operaciju ,fp-calc-2, tako da je registar B uvek pripremljen.

Dajemo i tabelu preostalih operacije

DODATNE OPERACIJE KALKULATORA

x , y	&09	na-lesgi	u=(x<=y)	u
x , y	&0A	na-gr=eq	u=(x >=y)	u
x , y	&0B	nas-neqi	u=(x <> y)	paredjenje u
x , y	&0C	na-grtr	u=(x > y)	brojeva u
x , y	&0D	na-less	u=(x < y)	u
x , y	&0E	nas-eqi	u=(x=y)	u
x$, y$	&11	str-[-eql	u=(x$<=y$)	,i
x$, y$	&12	str-gr-aq	u=(x$>=y$)	u
x$, y$	&13	strs-neql	u=(x$<>y$)	paredjenje u
x$, y$	&14	str-grtr	u=(x$ > y$)	nizava i
x$, y$	&15	str less	u=(x$ < y$)	,i
x$, y$	&16	strs-eql	u=(x$==y$)	u
x$	&1S	val$	u$=VAL$ x$, vrednavanja niza	,s
x$	&1D	val	u=VAL x$, vrednovanje niza	u

Jedina operaciji koju do sada nismo pomenuli je e to-p se kodom &3C ali se ona uopšte i ne može koristiti u okviru kalkulatora već se mora pozvati keo potprogram. Te operacije inače prevodi brojeva iz oblika , x E m u pokretni zaraz što nema za sada nije interesantno.

2 HL=HL*DE	CALL &30A9
GET_HL*DE	CALL &2AF4

Potprogram HL=HL DE obavlja množenje dva šesnaestobitne broja koji su na ulazu smešteni u HL i DE. Rezultat na izlazu je smešten u HL, registar DE je očuvan a BC se na koristi. Indikator prenose ,c, će biti setovan u slučaju prekoračenje opsega.

Potprogram GET_HL*DE radi isti posao, ali samo pod uslovom de je u toku izvršenje bejzik programa. U tazi sintaksna provera, povratak se vrši odmah. Osim toga, ako u toku množenje dode do prekoračenje, biće automatski pozvan prakid. 4 Out of memory

d) Pomoću naredba stk-data može se na stak dovesti **bilo koje konstanta**, direktno navodaći svaki bajt njanog binarnog zapisa. Tačnije, obavezno je navesti eksponent, e od bajtova mantise moze se dati jedan dva tri ili svin četiri pri čemu ce svi bajtovi koji nedostaju automatski biti zamenjeni nulama

Informaciju o tome koliko će bajtova za mantisu biti dato, nosi prvi bajt iza naredba stk-data , i to bitovi 7 i 6, se svojim kombinacijama 00 01 10 i 11 Bitovi od 0 do 5 služe za smeštanje eksponenta broja koji se šalja na stak Ako ja taj prostor nedovoljan bitovi 0 do 5 se svi postavljaju na nulu a eksponent dodaje posebno kao sledeći bajt U oba slučaja bilo da je eksponent ataο u prvi bajt bito da ja dodat kao samostalni bajt moraju se navsti još i bajtovi mantise u očekivanom broju

Da bi sva to bilo jasnija daćemo konkretan primar dovođanja na stek broja 43 217 Odgovarajući binarni zapis ja &86 &2C &DE &35 &40 što se jednostavno može dobiti i u bajziku stavljajuci prvo LET A=43 217, a zatim čitajući variablu A bajt po bajt.

Procedura pnprema bajtove koji ce sledili iza naredba , stk-data" ja sledeća

1) Redukujemo eksponent (prvi bajt binarnog zapisa) oduzimajući od njaga &50

redukovani eksponent=stvarni eksponent−&50

U našem slučaju eksponent iznosi &86 pa posla redukcija imamo &36

2) Odabiramo koliko bajtova za mantisu ćemo nevesti Taj broj naka bude n" Kod nas je n=4

3) Ako je redukovani eksponent manji od &40 onda se on može smestiti u šest bita u okviru prvog bajta iza stk-data Tada prvi bejt iznosi

b=(a−1)*64+ redukoveai eksponent

Ali ako je redukovani eksponent jednak &40 ili veći onda se dodaja kao posebni bajt dok prvi bajt iznosi samo

b=(a−1)*64

4) Konačno se daje , n bajtova za mantisu

U našem primeru, redukovani eksponent ja manji od &40 pe ja

b=3*64+&36=&Fa

Konačno konstanta 43 217 se dovodi na stak kao što je prikazano u primaru 35

```
35    DEFB &34          stk-data
      DEFB &Fa &2C &DE
      DEFB &35 &40
```

e) Operacija truncate nalazi celobrojni deo broja tj ono što se nalazi levo od dacimalna tačka To se pokiapa sa rezultatom funkcje INT samo ako je broj pozitivan

f) Operacija fp-calc-2 ja zapravo **zamena za bilo koju drugu**, pod uslovom da se u B registru nalazi stvarni kod operacija koju traba izvršiti Tako je recimo svejedno da li cemo postupiti prema primeru 36 ili 37

```
36    RST &28           FP_CALC
      DEFB &04          multiply
      DEFB &38          and-calc

37    LD B &04          pnprema za „multiply"
      RST &28           FP_CALC
      DEFB &38          fp-calc 2 konkretno multiply
      DEFB &38          and-calc
```

Upotreba operacije „fp-calc-2" je naročito značajna kada se u programu ne zna unapred koju operaciju treba izvršiti na primar ako se izračunava naki izraz koji je korisnik upravo ukucao Tada se potreban kod za operaciju ubacuje u B i izvršava se „fp-calc-2

g) Operacija tipa „series" su veoma značajna jer omogućuju izračunavanje funkcija pomocu redova U tu svrhu se, međutim ne koriste obični stepeni redovi već redovi **Čebišavljevih polinoma**, što obezbeduje brzu konvergenciju i veću tačnost Ako vas matematika previše ne interesuje možete preskočiti ovaj deo Ni mi nećemo mnogo ulaziti u detalje ali bismo želeli da vam barem damo ideju kako da sami konstite generator redova

Polinomi Čebisave se obično označavaju sa $T_n(x)$ gde je „x promanljiva a „n stepen polinoma (n=0 1 2 3) Prvih nekoliko polinoma su

$T_0(x)=1$
$T_1(x)=x$
$T_2(x)=2x^2-1$
$T_3(x)=4x^3-3x$
•
•
•

Svaki polinom se može izračunati na osnovu prethodna dva po rakurantnoj formuli

$T_n(x)=2xT_{n-1}(x)-T_{n-2}(x), \quad (n=2, 3, 4, \ldots)$

se prikazao sa BIN 0000101110111000 Prvi bajt (bajt veće težine) sadrži jednaaast odnosno BIN 00001011 a drugi 184 (BIN 10111000) tako da je

11×256+184=3000

Mikroprocesor Z80 međutim upisuja 16-bitna brojeva u memoriju obrnutim redom **prvo bajt manje tazine**! Dakle u slučaju broja 3000 prvi bajt će sadržati 184 a drugi 11 To moramo stalno imati u vidu ako iz memorije želimo da pročitamo naki 16-bitni broj. Na primer diva bajta na adresama 23635 i 23636 uvak sadrže adrseu na kojoj počinje bajzik program u memoriji Da bismo pročitali tu adresu, moramo izvršiti **PRINT PEEK 23635+256×PEEK 23636.**

Delako veći problam nsetaja kada treba zapisati naki **decimalan broj** Spektrum kao i većina drugih računara koristi tahniku bzv **pokretnog zareza**, i sva brojeva prikazuja u eksponencijalnom obliku **±mx2^e**, gde je „m" mantisa (uvak manja od jedinice, e veća ili jednaka 0 5) a „e" ja eksponent u rasponu između −127 i +127 Nija teško proveriti da ovakav način zapisivanja pokriva sva realne brojeve po modulu manje od oko 10^{+38} Pri toma svaki broj čiji je modul manji od oko 10^{-39} tretire se kao nula

Uzecamo kao primar broj 43 217, koji se može prikazati kao

43 217=0 875265625×2↑&

Mantisu 0 875265625 dalje ćemo prevesti u binarni oblik U binarnom brojnom sistemu takođa postoji decimalna tačka (možda bolje binarne tačka!) s tim što cifarska mesta iza tačka vreda redom 1/2 1/4 1/& itd umesto dakadnih 1/10 1/100 itd Tako ce recimo, dekadni broj 0 5 biti zapisen sa 0 1 binarno Mi se seda nećemo upuštati u detalja oko prevođanja brojeva iz jednog sistema u drugi Prosto ćemo deti konačni zapis mantise, zaokruzan na 32 cifre iza tacke

m=0 10101100110111100011010101000000

Na izgleda baš pragledno ali računar jedino uma da pamti jedinice i nula pa mu ni ovo neće predstavljati nikakav problam

„Spektrum" uvek odvaja **pet bajtova** za zapis brojeva u pokretnom zarezu U prvi bajt se upisuja eksponent „e uvećan za 128 a u preostala četin bajta (32 bita) smešta se mantisa, tačnija samo cifra iza decimalne tačke

Pošto je usvojeno da mantisa buda uvak veća ili jednaka 1/2, onda ja prvi bit iza decimalna tačka jedinica! Umesto ta jedinica, koje sa podrazumeva u taj bit se upisuja znak broja. nula za „+" a jedan za „−

Slika 3. Zapis broja u pokretnom zarezu



Naš broj 43 217 je pozitivan pa će prvi bit mantise biti nula Eksponent sabran sa 128 dace 134 (&86) pa tako konačno imamo zapis od pet bajtova &86 &2C &DE &35 &40

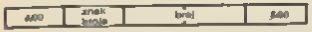
Sve ovo nija teško provariti ako u bajziku izvršimo na primar **LET a=43.217**, e onda pomoću funkcije **PEEK** pročitamo svaki bajt promanljiva „a Treba samo znati gde se u memoriji nataza bajzik promanljive a o toma ćemo govoriti kasnije

Što se tiče celih brojeva između −65535 i +65535 za njih „spektrum takođa rezerviše pet bajtova ali ja zapis drugačiji

- prvi bajt uvak je nula, što znači da sledi ceo broj
- drugi bajt sadrži nulu za pozitivna a 255 za negativna brojeve
- treći i četvrti bajt sadrže sem broj (prvo bajt manje težine) ali ako je broj negativan, dodaje mu se 65536, i
- peti bajt ja uvek nula

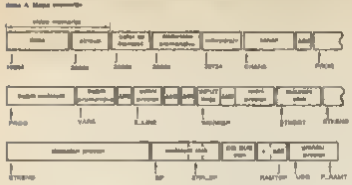
Zapis celih brojeva prikazan je na slici 3a e kao primar možemo uzeti broj −1 &00, &FF &FF &FF &00

Slika 3a. Zapis celog broja



Međutim prilikom obavljanja aritmetičkih operacija i celi brojevi se, po potrebi prevode u potpuni oblik pokretnog zareza Zapis nula ja pri toma jedinstven &00, &00 &00 &00 &00

Memorijska mapa



RAM (sl 4) je podeljen na više zasebnih blokova čiju funkciju ćemo detaljno objasniti! Neki blokovi imaju fiksnu početnu adresu ali većine ipak ne, pošto u toku rada dolazi do stalnog pomeranja i preraspodele prostora. Recimo unošenjem nove programske linije, svi blokovi osim od adrese PROG biće pomereni ka višim adresama, smanjujući tako slobodan prostor iza STKEND. Sema adresa PROG s druge strane zavisi od veličine bloka koji koristi mikrodrajv a takođe i od sadržaja kanalskih informacija. Ukoliko međutim nemamo mikrodrajv i ne diramo kanale, CHANS će uvek biti na 23734, a PROG na 23755.

Spektrum u svakom trenutku zna gde počinje koji blok, jer se sve adrese čuvaju u posebnom prostoru, tzv **sistemskih varijabli**, o čemu ćemo još govoriti.

VIDEO MEMORIJA

1. Spektrumov ekran sastoji se od 49152 tačke: 256 po horizontali i 192 po vertikali, a sve slova, cifre i ostali znakovi ispisuju se u poljima veličine 8×8. Takvih polja ima ukupno 32×24. Svaka tačka u okviru jednog polja može imati samo jednu od dve boje, ili će to biti boje podloge (PAPER) ili boja teksta (INK), dok polje za sebe može imati i pojačan sjaj (BRIGHT) i može i treptati tj. naizmenično menjati boju podloge i teksta (FLASH). Sve te informacije o slici zapisane su u prvih 6912 bajtova RAM-a odakle ih „spektrum" čita i šalje na TV prijemnik.

Tačka na ekranu predstavljena je **jednim bitom** u memoriji. Ako je tačka boje podloge odgovarajući bit je resetovan (nula), a ako je tačka boje teksta odgovarajući bit je setovan (jedinica). Osam tačaka, preme tome, čine jeden bajt. Tih bajtova ukupno ima 6144 i redaju se počev od adrese 16384.

Jednom bajtu odgovara na ekranu horizontalna crtica dužine osam tačaka. Ako odgovarajući bajt sadrži BIN 11111111 crtica je neprekidne, jer sve tačke imaju boju teksta.

Prvi bajt video memorije predstavlje crticu u gornjem levom uglu ekrana, što jednostavno možemo proveriti sa **POKE 16384,255**. Međutim, raspored ostalih bajtova nije baš uvek onakav kakav bismo očekivali. Najbolje je umesto bilo kakvog objašnjenja izvršiti kratak bajzlik program, koji redom popunjava video bajtove u memoriji i onda pratiti šta se dešava na ekranu. Stvar će se ubrzati izbacivanjem pauze sa linije 30. Druga pauza, na liniji 50, samo služi da spreči ispisivanje raporta, jer bit to inače odmah izbrisalo dve donja reda.

```
1  10 FOR n=16384 TO 22527          10 FOR n=22528 TO 23295
   20 POKE n, 255                   20 POKE n, 0
   30 PAUSE 10                      30 PAUSE 10
   40 NEXT n                        40 NEXT n
   50 PAUSE 0                       50 PAUSE 0
```

Imajući sve u vidu, izgleda da je vrlo komplikovano pronaći u memoriji pravi bajt a zatim i odgovarajući bit u okviru tog bajta, na osnovu koordinata x i y neke tačke. Ne sreću, oko toga ne moramo mnogo brinuti, jer u ROM-u postoji jedan veoma elegantan mašinski potprogram koji obavlja taj zapetljan proračun. Nije međutim loša vežbe napraviti bajzlik potprogram koji simulira naredbu **PLOT** isključivo korišćenjem **POKE**.

2. Preostalih 768 bajtova video memorije, počev od 22528, nose informaciju o boji za svako znakovno polje. To su tzv. **atributi**.

Od osam bita, sa kojiko sveki atribut raspolaže, najmiže tri rezervisane su za boju teksta između INK 0 i INK 7. Sledeća tri nose boju podloge (PAPER 0 — PAPER 7), bit 6 sadrži informaciju o sjaju (BRIGHT 0 ili BRIGHT 1), a bit 7 informaciju o treptanju (FLASH 0 ili FLASH 1). Šematski prikaz dat je na slici 5. Redosled atributa u memoriji odgovara redosledu polja na ekranu, onako kako sa imée ispisuju.

e) Operacije „jump-true" „jump" i „dec-jr-nz" koriste se za grananje u procesu računanja. Iza sveke od tih naredbi tj. iza odgovarajućeg koda &00, &33 ili &35 mora se navesti još jeden bajt, koji daje **relativne rastojanje odredišta** na koje će se skočiti. Slično mašinskoj naredbi JR.

Operacija „jump-true" je uslovni skok, koji se izvršava samo ako je poslednji rezultat na steku bio logički istinit tj. različit od nule. U protivnom se skok ne vrši, već se izvršava sledeća operacija. U oba slučaja obavlja se **brisanje poslednjeg rezultate** se steka, kao da je izvršena operacija „delete". Kao primer navodimo situaciju u kojoj se testira broj x: ako je X>0, skok se vrši na kraj računanja, a eko nije, obavlje se pre izleska još nekoliko operacija, da bi krajnji rezultat bio x*PI.

```
32    RST &28        FP_CALC          x
      DEFB &31       duplicate        x, x
      DEFB &37       greater-0        x, (0 ili 1)
      DEFB &00       jump-true        x
      DEFB $05       to END           x
      DEFB &A3       stk-pi/2         x, PI/2
      DEFB &31       duplicate        x, PI/2, PI/2
      DEFB &0F       addition         x, PI
      DEFB &04       multimply        x*PI
END   DEFB &38       end-calc         x ili x*PI
```

U ovom primeru relativne vrednost skoka iznosi &05 e to je u stvari razlika između adrese odredišta i adrese samog bajta ze vrednost skoka. Nema nikakve potrebe da tu razliku računamo, već sve treba prepustiti asembleru. Teko bismo umesto DEFB &05, ukucali

```
      DEFB  END — $
```

Pri tome je sasvim svejedno da li se skok obavlje unapred, ili unazad.

Naredba „jump" radi slično kao i „jump-true", s tom razlikom što se skok ne zadeto odredišta obavlje bezuslovno, a računski stek ostaje naizmenjen.

Konačno operacija „dec-jr-nz" služi za formiranje programskog ciklusa u okviru procesa računanja, uz kontrolu broja ponavljanja. Da bi se ove naredba uopšte mogle konstitui pre poziva kalkulatora se u registar B mora ubaciti broj ciklusa koji će biti izvršeni u toku računanja. Tej broj će, čim se izvrši instrukcije RST &28 automatski biti pranet u sistemsku varijablu BREG i odatle će sa kasnije koristiti.

Kada računar naide na naredbu „dec-jr-nz" on umanjuje BREG za jedinicu i obavlje skok na naznačeno odredište samo pod uslovom da brojač još nije stigao do nule — potpuno slično mašinskoj naredbi DJNZ.

Primer koji dajemo predstavlja ciklus sa ukupno sedam prolaza, pri čemu se u svekom prolazu broj „n" na staku sabira se jedinicom. Na ulazu je n=1, a na izlazu n=8.

```
33    LD B 7           sedam ciklusa
      RST &28          FP_CALC
      DEFB $A1         stk-one          1
LOOP  DEFB &A1         stk-one          n, 1
      DEFB &0F         addition         n=n+1
      DEFB &35         dec-jr-nz        n
      DEFB LOOP — $    to LOOP          n
      DEFB &38         end-calc         8
```

Unutar ovakvih ciklusa **ne smeju se naći** operacije koje i same koriste BREG za svoje potrebe (generator redova i sve funkcije koje se računaju preko redova).

b) Operacija „x mod-y" ne postoji u bajziku, ali može biti veoma korisna. Ona obavlje deljenje dva broja a u rezultatu deje istovremeno i ceo deo količnike v=INT(x/y) i ostatek u=x−y*v.

c) Operacije „stk-zero", „stk-one", „stk-half", „stk-pi/2" i „stk-ten" služe da bi se ne stek dovele neke često korišćene konstante: 0, 1, 0.5, PI/2 i 10. Poneked se na osnovu tih konstanti mogu izvseti i druge, kao recimo PI.

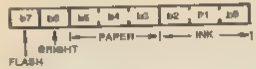
```
34    DEFB &A3         stk-pi/2         PI/2
      DEFB &31         duplicate        PI/2, PI/2
      DEFB &0F         addition         PI
```

tepala 2:

OPERACIJE KALKULATORA

	&00	jump-true	skak ako je "x" istinite	-	
v	&01	exchanga	zamena mesta	y , x	
	&02	delete	Brisanje sa staka	-	
v	&03	subtract	u=x-y, oduzimanje	u	
v	&04	multiply	u=x*y, umnožanje	u	
v	&05	division	u=x/y, deljenje	u	
v	&06	to-power	u=x^y, stepenovanje	u	
, y	&07	or	u=x OR y, Broj OR Broj	u	
x , y	&08	na-and-na	u=x AND y, broj AND broj	u	
x , y	&0F	addition	u=x+y, sabiranje	u	
n0, y	&16	str-and-ns	u=n$ AND y, niz AND broj	u$	
n$, y$	&17	str-cat	u$=n$+y$, nadovezivanje nizova	u$	
x$	&19	usr-$	u=USR n$, adresa grafičkih znakova	u	
v	&1A	read-in	čitanje podatka u$ sa stroja "x"	u$	
x	&1E	negate	u=-x, promena znaka	u	
x$	&1C	case	u=CODE x$, ASCII kodiranje	u	
n$	&1E	len	u=LEN x$, dužina niza	u	
x	&1F	sin	u=SIN x, sinus	u	
v	&20	cos	u=COS x, kosinus	u	
	&21	tan	u=TAN x, tangens	u	
	&22	asn	u=ASN x, arkus sinus	u	
	&23	acs	u=ACS x, arkus kosinus	u	
	&2a	atn	u=ATN x, arkus tangens	u	
	&25	ln	u=LN x, prirodni logaritam	u	
e	&26	exp	u=EXP x, eksponencijalna funkcija	u	
	&27	int	u=INT x, najmanji ceo broj	u	
	&28	sqr	u=SQR x, kvadratni koren	u	
	&29	sgn	u=SGN x, znak broja	u	
	&2A	abs	u=ABS x, apsolutna vrednost	u	
	&2B	peek	u=PEEK x, očitavanje sadrzaja memorije "x"	u	
	&2C	in	u=IN x, čitanje porta "x"	u	
	&2D	usr-no	u=USR x, poziv mašinskog programa	u	
	&2E	str$	u$=STR$ x, pretvaranje broja u niz	u$	
	&2F	chr$	u$=CHR$ x, ASCII zakodiranje	u$	
	&58	not	u=NOT x, logička negacija	u	
	&31	duplicate	čuvanje poslednjeg broja	x , x	
	&32	x-mod-y	u=x MOD y, v=x DIV y	u , v	
	&33	jump	Bezuslovni skok	x	
	&34	stk-data	dovlačenje podatka "u" na stak	x , u	
	&35	dec-jr-nz	kontrola ciklusa (kao DJNZ)	k	
	&36	less-0	u=x<0, poređenje sa nulom	u	
	&37	greater-0	u=x>0, poređenje sa nulom	u	
	&38	end-calc	završetak računanja	x	
	&39	get-argt	priprema argumenta za SIN i COS	u	
	&3A	truncate	celobrojni deo	u	
	&3B	fp-calc-2	izvršenje aparacije sa kodom u $	?	
	&30	re-stack	preuzeo broja u pokretni zarez	x	
	&81	series-01		u	
	"	"	generator reseva	"	
	"	"	čapljavljivih palindma	"	
	&9F	series-1F		u	
	&A0	stk-zero		x , 0	
	&A1	stk-one		x , 1	
	&A2	stk-half	osvajanje konstanta na stek	x , 1/2	
	&A3	stk-pi/2		x , PI/2	
	&A4	stk-ten		x , 10	
	&C0	st-mem-0		x	
			smeštanje broja		
			u memoriju		
	&C5	st-mem-5		x	
	&E0	get-mem-0		x	u
	"	"	uzimanje broja		
			iz memorije		
	&E5	get-mem-5		u	

Slika 6. Atribut



Bilo kakva promena atribute uopšte neće uticati na adržaj slike, vec jedino na boju. Recimo možemo napisati na ekranu neki tekst a onde ubaciti u sve atribute vrednost BIN 00111111 pomoću naredbe **POKE**, naravno. Time će ekran postati potpuno prazan (FLASH 0 BRIGHT 0 PAPER 7 INK 7). Medutim ako vrsimo atribute ne BIN 00111000 atika će se ponovo pojaviti bez ikakvih izmena (INK 0)

BAFER ZA STAMPAC

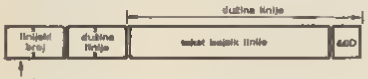
Blok od 256 bajtova, sa početkom na 23296 naziva se bafer za štampač. Tu se formira tekst koji se štampa i tek onda se šalje na printer. Inače, u drugim prilikama možemo ove bajtove koristiti i za svoje potrebe

BEJZIK PROGRAM

Sve linije bejzik programe čuvaju se u memoriji jedna iza druge, onim redom kojim se pojavljuju i kada ih izlistamo na ekranu. Prva dva bajta u zapisu svake linije sadrže linijski broj pri čemu prvo dolazi bajt **veće težine**! To je jedini slucaj takvog prikazivanja brojeva na spektrumu i treba ga dobro upamtiti.

Iza linijskog broja slede dva bajta koja sadrže dužinu linije (tekst+ENTER), u normalnom poretku prvo bajt manje težine. Konačno dolazi tekst same linije, označen ne kreju sa ENTER (&0D) kao što je prikazano na slici 6

Slika 6. Bejzik linija



Tekst se sastoji od niza naredbi i drugih znakova pri čemu jednoj naredbi ili znaku odgovara jedan bajt, sa odgovarajućim ASCII kodom. Iza svake numeričke konstante koja se u liniji pojavljuje umetnut je bajt &0E praćen sa pet bajtova binarnog zapisa konstante u pokretnom zarezu. Na slici 7 je prikazan kompletan zapis bejzik linije 10 GO TO 3000 što obuhvata ukupno 16 bajtova

Slika 7. Primer bejzik linije



Šest bajtova sa ubacuje i u liniju sa naredbom DEF FN iza svakog argumenta navedenog unutar zagrade. Prvi bajt je opet &0E a ostalih pet je samo rezervisano za stvarnu vrednost argumenata koji ce tu biti upisan u fazi izračunavanje same funkcije. Na slici 6 dajemo primer linije 20 DEF FN p(x, g$)=x+LEN g$

Slika 9. Primer bajtik slike



BEJZIK PROMENLJIVE

Promenljiva sa smeštaju u blok memorija počev od adrese VARS onim redom kojim su i definisane. Spektrum prepoznaje šest različitih tipova varijabli. Odgovarajući tip sadržan je u prvom bajtu zapisa promenljiva (bitovi 7 & 5) dok preostali bitovi 0–4 sadrže ime promenljive umanjeno za &60

Slika 8. Prvi bajt varijable



Redukovano ime uvek je u opsegu između &01 i &1A, što odgovara stvarnom imenu između a i, z (&61–&7A). Ako ima sadrži više od jednog slova, redukuje se samo prvo

1 Numerička varijabla čija je ime jedno slovo, pokazuje se sa šest bajtova. Prvi nosi oznaku tipa 011 i redukovano ime a ostalih pet sačinjavaju vrednost sama promenljive

Slika 10. Numerička promenljiva



2 Numerička varijabla sa dužim imenom ima tip 101. Prvo slovo imena se redukuje i ubacuje u prvi bajt zapisa, ostala slova sa navoda redom bez ikakvih izmena, a poslednje slovo je obavezno dato sa setovanim sedmim bitom (stvarni kôd je sabran sa 128) iza imena dolaze pet bajtova za sam broj

Slika 11. Numerička promenljiva sa dužim imenom



3 Numerički niz ima tip 100 a ima ja uvek jedno slovo. Dva bajta rezervisana su za dužinu cela varijable zatim sledi jedan bajt koji sadrži ukupan broj dimenzija, a onda redom dolaze maksimalne vrednosti svih indeksa po dva bajta svaki. Konačno imamo elemente niza od po pet bajtova takvim redosledom da se najbrže menja poslednji a najsporija prvi indeks. Dakle prvo dolaze elementi koji imaju zajednički prvi indeks, a od njih prvo oni koji imaju zajednički drugi indeks itd

Slika 12. Numerički niz



4 Dimenzionisan znakovni niz (tip 110) sasvim je sličan numeričkom nizu samo što elemanti zauzimaju po jedan bajt.

KALKULATOR

Već smo ranije rekli da pre bilo kakvog računanja, argumenti moraju biti postavljeni na računski stek. Taj posao svakako obavljaju posebni potprogrami iz ROM-a

Pretpostavimo da je u početku stek prazan (to znači da je STKBOT=STKEND), a mi želimo da tamo ubacimo broj 85. Sav posao se sastoji u tome što broj 85 treba smestiti u akumulator a onda pozvati rutinu koja šalje akumulator na stek. U rezultatu toga imaćemo na računskom steku pet bajtova, počev od STKBOT u koje je upisan broj 85. STKEND pokazuje odmah iza tih pet bajtova

Dalje želimo na stek da pošaljemo broj 18. Opet ćemo pripremiti akumulator i pozvati pomenuti potprogram. Broj 19 će doći odmah iza (ili iznad) prethodnog broja 85 a ceo stek sada ima deset bajtova. Svaki sledeći broj obavezno dolazi iznad ostalih

Spektrum uvek obavlja operacija sa brojevima u samom vrhu računskog steka. Ako mu naredimo da obavi operaciju Kosinus on će u stvari izračunati kosinus poslednjeg broja na računskom steku. Sam taj broj će odmah biti uklonjen sa steka, a umesto njega će doći rezultat izvršena operacija. Potpuno isto rade i funkcija SIN, TAN, LN, EXP, ABS, INT itd. Sve su to primeri tzv unarnih operacija. One deluju na poslednji broj na steku i zamenjuju ga rezultatom

Postoje i tzv binarna operacija. Deluju na dva broja, da kažemo, x i y, pri čemu je y u vrhu steka a x odmah pod njim. Po završenoj operaciji oba broja nestaju sa steka, a ostaje samo rezultat. Primer binarnih operacija su sabiranje, oduzimanje, množenje, deljenje, stepenovanje itd

To je za sada sve što se tiče brojava i operacija sa njima. Međutim, spektrum uma da radi i sa nizovima slova i znakova. Razuma se i nizovi sa kojima se obavljaju neke operacija moraju doći na računski stek. Pri tome da se stvari ne bi previše komplikovala, svaki niz na steku takođe zauzima pet bajtova, kao da je u pitanju broj. U tih pet bajtova, međutim, nije zapisan sam niz već parametri koji ga potpuno određuju. Niz sa inače nalazi negde u memoriji (obično u rednom prostoru) a na steku se čuva njegova adresa (drugi i treći bajt) i dužina (četvrti i peti bajt). Prvi bajt, konsti samo naredba LET a nama uopšte nije važan. Obično ga stavljamo na nulu. Što se tiče prenošenja parametara niza na stek o tome opet ne treba brinuti, postoje i tu posebni potprogrami. Treba samo upamtiti da spektrum uopšte na pravi razliku između brojeva i nizova na steku. Rečimo bez obzira na to šta zaista sadrže poslednjih pet bajtova na steku, podjednako uspešno će se izvršiti i operacija **COS** i operacija **LEN**, jer će svaka od njih tretirati onih pet bajtova onako kako joj odgovara

FP_CALC — RST &28

A Potprogram FP_CALC poziva se kada treba obaviti bilo kakvo računanje. Ulaznih parametara nema, a podrazumeva se da su prethodno svi brojevi sa kojima se računa preneti na računski stek u očekivanom redosledu

Iza naredbe **RST &28** mora doći niz bajtova koji datinišu računska operacije onim redom kojim će se izvršavati. Svaka operacija ima svoj kôd i predstavljena je jednim bajtom. Na kraju tog niza obavezno dolazi bajt &38 kao znak da je računanje završeno

Izlaznih parametara nema registri nisu očuvani, a rezultat računanja ostaće na računskom steku kao poslednja vrednost

Primer 31 pokazuje niz aritmetičkih operacija.

```
31    RST  &28        FP_CALC
      DEFB &1F        sinus
      DEFB &0F        sabiranje
      DEFB &27        INT
      DEFB &04        . množenja
      DEFB &0F        sabiranje
      DEFB &38        . kraj
```

Daćemo tabelu svih operacija ugrađenih u spektrumov kalkulator. Većina njih je naposredno prisutna i u bajziku, dok su druga dostupna samo u mašinskom jeziku

Svaka operacija (tabela 2) ima svoj heksadekadni kod i skraćeni naziv. Zatim sledi kratko objašnjenje same operacija i konačno stanja računskog steka po njenom izvršenju. Prva kolona tabela prikazuje računski stak na ulazu u operaciju

Objašnjenja rada nekih specijalnih operacija dato je i posebno van tabela

poslednjeg bajta te linije (ENTER) nalazice se u registru BC Ali eko je pretraživanje počelo iz sredine neke linije i pri tome je naredba E otkrivena u istoj toj linij onda varijable NEWPPC i registar BC na izlazu iz LOOK_PROG imaju nepromenjenu vrednost sa ulaza.

U svim slučajevima slično kao kod EACH_STMT akumulator ne izlazu sadrži karakter sa adrese HL

Kao primer ćemo napraviti program koji pronalazi sve **GO TO** naredbe da bi ih na neki način obradio (recimo da se radi o pranumeraciji) Samu tu obradu označili smo uopšteno sa MODIFY a podrazumeva se da registar HL pri tome ostaje očuvan

Pretraživanje počinje od prve bajtzik linije (PROG) a registar E sadrzi potreban kôd &EC Poziva se LOOK_PROG i odman obavlja povratak eko je traganje završeno (RET C) EACH_STMT se koristi da bi se prešlo pred sledecu naredbu

```
29    LD          HL  (PROG)
      DEC         HL
LOOP  LD          DE  &00EC
      CALL        LOOK_PROG
      RET         C
      CALL        MODIFY
      DEC         HL
      LD          DE  &0200
      CALL        EACH_STMT
      JR          LOOP
```

14 LOOK_VARS CALL &29B2

Potprogram LOOK_VARS pronalazi zadatu bajzik promenljivu u memoriji li nalazi adresu odgovarajućeg argumenta u DEF FN naredbi ukoliko je upravo u toku izracunavanje neke DEF FN funkcije

Ulaznih parametara nema ali sistemska varijabla CH_ADD mora sadržeti adresu na kojoj počinje tekst sa imenom promenljive Obično se to ime nalazi u okviru nekog aritmetičkog izraza u bejzik programu ali mi možemo formirati isti tekst i bilo gde na drugom mestu Jedino moramo voditi računa o tome da u slučaju dimenzionisanih nizova, iza svakog indeksa **mora stajati** odgovarajuci binarni zapis u pokretnom zarezu (oznaka &0E plus pet bajtova za sam broj) Ukoliko tekst nema smisla tj ne predstavlja ime bilo kakve promenljive usledice prekid se greškom C Nonsense in BASIC

Na izlazu iz rutine LOOK_VARS indikator prenosa_C je satovan ukoliko varijable nije pronađena. Teda HL sadrži adresu imena varijable (kao CH_ADD na ulazu) a indikator nule ‚Z' setovan je samo eko se tražila promenljiva tipe dimenzionisanog niza

Ukoliko je na izlazu C=0 to znači da je promenljiva pronađana i tada HL sadrzi odgovarajucu adresu Ako se ime promenljive sastoji iz više slova HL ce pokazivati poslednji bajt imena Dakle u bilo kom slučaju naredbom INC HL se prelazi iza imena varijable Indikator nule ‚Z' je resetovan samo ako je promenljiva običnog numeričkog tipa uključujući i FOR_NEXT Za bilo koju vrstu niza je Z=1

Ostali registri nisu očuveni

Primecujemo da u slučaju dimenzionisanih nizova ne izlazu ne dobijemo adresu trezenog elementa već adresu niza kao celine Izdvajanje naznecenog elementa moguce je primenom rutine STK_VAR

15 STK_VAR CALL &2996

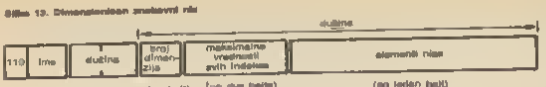
Potprogram STK_VAR može da obavlja dva potpuno različite posla Prvi je kao sto smo vec napomenuli pronalaženje pojedinačnih elemenata numeričkog niza a drugi je smestanje na računski stek parametara nekog alfanumeričkog niza. O ovoj drugoj funkciji cemo posebno govoriti u okviru kalkulatora a sada se samo zadržavamo na prvoj

Ulazni parametri u STK_VAR su isti oni koje ne izlazu daje LOOK_VARS Zapravo STK_VAR se uvak i zove neposredno iza LOOK_VARS ali samo pod uslovom C=0 (varijabla je pronađena) i Z=1 (varijable je tipa niza

```
30    CALL        LOOK_VARS
      JP          C  REPORT_2
      CALL        Z  STK_VAR
```

Sa REPORT_2 smo oznacili adresu na kojoj se poziva raport_2 Variable not found a skok se obavlja samo ako je C=1 Za C=0 i Z=1 poziva se STK_VAR.

Na izlazu ce HL sadrzati adresu neposredno ispred zadatog elementa numeričkog niza a BC će pokazivati bajt ispred prvog elementa niza Dakle posle INC HL pomericemo se na prvi od pet bajtova elementa niza koji smo tražili DE na izlazu uvek sedrži &0005 a akumulator je nula



5 Običan znakovni niz (string) ime tip 010 Dva bajta sadrže dužinu niza, odnosno broj znakova. Ukoliko je niz prazan, onde je dužina nula I teksta nema. Ineče, sledi tekst se svim znakovima



6 Kontrolna varijable za FOR–NEXT petlje (tip 111) zauzima ukupno 19 bajtova Po pet bajtova je razervisano za tranutnu vrednost promenljive graničnu vrednost u petlji i korak sa kojim se promenljiva menja Od tn poslednja bajta prve dva sadrže linijski broj petlje odnosno linije sa naredbom FOR pri čemu prvo ide bajt manje težine! Poslednji bajt je redni broj naredbe u linij na kojoj počinje petlja, a to je uvek za jeden veće od rednog broja naredbe FOR



EDITNI PROSTOR

Iza bloka bejzik varijabli nalazi se jeden graničnik u vidu bajta &80 koj računaru samo govori da promenljivih više nema Iza tog bajta počinje tzv editni prostor koji služi za prihvatanje programske linije koja se trenutno ukucava, ili koja se poziva pomoću komande **EDIT.**

Dok kucamo liniju ili je prepravljamo, one je u editnom prostoru zapisana prosto kao niz znakova onako kako je vidimo u dnu ekrane (neme binarnog zapisa linijskog broja, niti bilo kekvih umetnutih bajtova) Tek kada pritisnemo **ENTER**, računar će proveriti sintaksu dovesti liniju na format koji smo ranije opisali i preneti je u blok se bejzik programom ili je neposredno iz editnog prostora izvršiti eko sa radi o direktnoj naradbi

Poslednji bajt u editnom prostoru je uvak **ENTER**, e za njim sledi graničnik &80

RADNI PROSTOR

Na adresi WORKSP pocinja radni prostor računare Prvi deo je razervisan za liniju koje se unosi u toku izvršenja naredbe **INPUT**, a drugi iza oznake &0D konsti se u različitim situacijame keda privremeno treba smestiti neke podatke radi njinove obrade Tu recimo dolazi zaglavlje programa koji se uzima sa kasete ili nizovi znakova se kojima se vrše računske operacije Za vreme naredba **MERGE**, čitav program se kasete će biti unet u radni prostor da bi odatle bio prenet na svoje pravo mesto, kombinujući sa sa već postojećim programom u bejzik bloku

Radni prostor neme fiksnu veličinu, već se stvare po potrabi Kada se na konsti sadrži

samo bajt &0D. U ROM-u kako sto cemo kasnije videti postoje gotovi potprogramt za stvaranje i oslobadjanje ladnog prostora sto i sami mozemo koristiti

RAČUNSKI STEK

Sve računske operacije spektrum iskljucivo obavlje sa brojevima koji se nalaze ne tzv računskom steku To je blok memorija čiji početak (ili dno) i kraj (odnosno vrh) su odredeni adresame STKBOT i STKEND Brojevi ne staku su poredani jedan iza drugog (ili jeden iznad drugog) i sveki zauzima pet bajtova.

Postoje posebne rutine u ROM-u koje smeštaju brojeve na računski stek ili ih odende uzimaju Svaki nov broj dolazi na sam vrh steka zatrpavajuci ostale I mi ako želimo da nesto računamo u mašinskom jeziku, moramo svoje brojeve poslagati ne računski stek, o čemu cemo detaljno govoriti kasnije

SLOBODAN PROSTOR

Iznad računskog steka počevši od adrese STKEND, prostire se slobodan prostor odnosno neiskorišćeni bajtovi koji čine preostalu memoriju dostupnu bejziku Kada se bilo koji od već pomenutih memorijskih blokove poveca, to ide na račun smanjenje slobodnog bloka Medutim čim slobodnog prostora nema dovoljno računar prekide rad i daje izveštaj 4 Out of memory '

MAŠINSKI STEK

Prvo na šta nailazimo ize slobodnog prostora jeste maštnski stek procesora Z80 Tu su smeštene adrese za povratak iz mašinskih potprograma a privremeno se čuvaju i razni drugi podaci Za razliku od računskog steka mašinski raste **nenize,** ke menjim adresama pa zato prvo nailazimo na vrh staka, koji je ineče odreden sadržajem procesorskog registra SP

Dve bajta u samom dnu mašinskog staka sadrže adresu &1303 što je u stvari rutine za štampanje raporta iz ROM-a Bajt manje težine je označen adresom ERR_SP kojom će ineče biti napunjen register SP u slučaju bilo kakve greške (efekat brisanja mašinskog staka) O tome cemo kasnije još govoriti

Ispod mašinskog steka (po adresame iznad) je prostor koji koristi bejzik neredba **GO SUB,** pa se i zove GO SUB stek Pri svakom pozivu potprograma iz bejzika, ne stek se postavljaju tri bajta linijski broj (dva bajta) i redni broj neredba na koju će se izvršiti povratak (to je odmah iza **GO SUB**) Neredba **RETURN** ce prosto skinuti sa steka poslednja tri bajta i na osnovu njih obaviti povratak iz potprograme Medutim ako pra toga nije izvršene ni jedna **GO SUB** neredba, i stak je prazan Imaćemo grešku 7 RETURN without GO SUB

Kako uopšte računar može znati de li je OO SUB stek prazen? Jednostavno u samom dnu steka se nalaze dva granična bajta Prvi nije određen a drugi obevazno sadrži &3E Ako neredba **RETURN** pokupi sa steka ova dva bajta umesto linijskog broja to će odmah biti jasno jer ni jedan linijski broj ne može imati bajt veće težine jednak &3E!

Adresa bajta &3E označeve sa se RAMTOP i to je poslednje adresa RAM-a dostupna bejziku Sve što je iza toga bezbedno je ćak toliko de će biti pošteđeno i od neredbe NEW

Obično se RAMTOP nalazi blizu stvernog kreja RAM-a (P_RAMT) teko de bezbedni deo memorije zauzima samo 168 bajtova, predviđenih za grafičke znekove (21 znak po osam bajtove) Otuda se odmah iza RAMTOP a nalazi adresa UDG (User Defined Graphics)

RAMTOP mozemo postaviti i sami na neku nižu adresu Na primer posle **CLEAR 30000,** RAMTOP ce biti na 30000 a čitav prostor počev od 30001 osloboden je za nase potrebe

KANALI

Komunikacija se nekim peritenjskim uredajem bilo de je u pitanju tastatura štampač ili TV ekran obavlje se preko tzv **kanele i struje** Ne primar de li će pri štempenju nekog teksta informacije otići na gornji deo ekrane_ odnosno tamo gde normelno radi neredbe

10 NUMERIC — CALL &2D1B

Pomoću ove rutine ispituje se da li neki karakter predstevje cifru dekadnog sistema Na ulazu akumulator sadrzi spitiveni kafakter na izlazu su svi registri uključujuci i A očuvan a indikator prenosa C je resetovan samo ako se radi o cifri

11 ALPHANUM — CALL &2C8B

Potprogram ALPHANUM na ulazu ima nek karakter u akumulatoru a na izlazu ce indikator prenosa C biti setoven samo ako se radi o slovu ili cifri Svi registri uključujuci i A su sačuvani

12. EACH_STMT — CALL &1988

Potprogram EACH_STMT služi za nalaženje adrese proizvoljne neredbe u okviru jedne bejzik linije Na ulazu HL sadrži polaznu adresu ze pretraživenje To će ako sa pretrazuje cela linija biti adresa neposredno ispred prve neredbe (nikako ne adresa linijskog broje!) III ako se polazi iz sred ne linija bice to adresa znaka Dakle u bilo kom slučeju HL na ulazu trsba postaviti neposredno ispred neke naredbe

Neredba koja se traži može biti zadata na dve načina iti čamo u E registar ubaciti odgovarajući kôd ili u D registar odgovarajuci redni broj meran počev od naredbe ispred koje je i postavljen HL Da ne bi bilo nesporazume najbolje je ubeciti nulu u onej od ove dva ragistra (D ili E) koji ne koristimo

Kao primer trazicemo adresu sedme naredbe po redu u liniji 7000 Prvo nalazimo adresu te linije i pomeramo se pred sam tekst Registar E ne koristimo pe čemo ga posteviti ne nulu

```
27    LD      HL 7000
      CALL    LINE_ADDR
      INC     HL
      INC     HL
      INC     HL
      LD      DE &0700
      CALL    EACH_STMT
```

Povretak iz EACH_STMT uslediče bilo de je pronađene naredba se kodom E bilo de je pronađene naredba sa rednim brojem D bilo de je dostignut kraj linije (ENTER)

Na izlazu ce indikator prenosa C biti resetovan samo ako je neradba pronađena U protivnom je C=1 dostignut je kraj linije i HL sadrži adresu karaktera ENTER

Ako je pronadena neredba E indikator nule Z biče resetoven a HL sadrži adresu same neradbe Medutim ako je pronađena naredbe D biče Z=1 a HL pokazuje neposredno ispred neredbe odnosno neposredno iza prethodne (znak ':')

U svim slučajevime registar E ne izlazu ostaje očuven registar B se ne koristi a A sadrži bajt sa adrese HL

Često se EACH_STMT poziva de bi se preskočila neka naredba odnosno da bi se došlo pred sledecu U tom slučeju ako je adresa naredbe koja sa preskače smeštene u HL trebe postupiti kao u primeru

```
28.   DEC     HL
      LD      DE, &0200
      CALL    &1988
```

13. LOOK_PROG — CALL &1D05

Rutina LOOK_PROG koristi sa za pretraživenje bejzik programa u cilju nelazenja neka zadate neredbe Automatski se ispituje linije po linija uzestopnim pozivanjem EACH_STMT

Adresa od koje počinje traganje nalazi se u registru HL na izlazu Slično kao u slučaju EACH_STMT mora se uzeti adresa **ispred neke neredbe,** tečnije adresa znaka Medutim ako se počinje od početka neke linije uzima se adresa **neposredno ispred tinijskog broje,** a ne ispred prve naredbe kao kod EACH_STMT Obično se ispred linijskog broja nalaz karekter ENTER kao kraj prethodne linije ili oznaka &80 (početak bejzik bloka)

Naredba koja se traži smesta se u registar E Nejsigurnije je uvek pri tome staviti i nulu u D meda je to samo neophodno ako pratražívanje počinje iz sredine linije

Ne izlazu će indikator prenosa C biti setoven ukolliko u čitavom programu počev od zadate adrese nije pronađena ni jedna neredba sa kodom E U tom slučeju HL sadrži adresu iza programe (VARS) a indikator nule Z je takođe satovan

Medutim eko je C=0 to znači da je neredba E pronadena Registar HL sadrži njenu adresu sistemska verijable NEWPPC sadrži linijski broj odgovarajuće bejzik linije a adresa

kodom većim od &20 uz izuzetak znaka &0D (ENTER) Svi nevažeći karakteri se ignorišu pri interpretaciji bejzike (prazne polje komandni simboli za boju itd) Ukoliko ne ulazu u GET_CHAR varijabla CH_ADD sadrži adresu nekog nevažećeg znaka pozicije se pomera unapred za jedno mesto i traži se **prvi važeći** karakter Tek kada se on pronađe obavlje se povratak uz odgovarajući sadržaj registara A i HL kao i varijable CH_ADD

Primer 25 će dati prvi važeći bejzik karakter linije u editnom prostoru

```
25    LD      HL (E_LINE)
      LD      (CH_ADD) HL
      RST     &18
```

Ukoliko želimo da se iz našeg mašinskog programa uspešno vratimo u bejzik, moramo obezbediti da CH_ADD pri povratku ima **istu vrednost kao i na početku programa** A taj uslov neće biti ispunjen ako koristimo rutine kao što je GET_CHAR Najbolje je, zato, ne samom početku našeg programa staviti

```
      LD      HL (CH_ADD)
      PUSH    HL
```

a na kraju neposredno ispred RET

```
      POP     HL
      LD      (CH_ADD) HL
```

6. NEXT_CHAR RST &20

Potprogram **NEXT_CHAR** je praktično isto što i GET_CHAR samo što pre početka rada uvećave CH_ADD za jedinicu Prama toma, na izlazu se neće dobiti prvi važeći karakter već sledeći važeći karakter Uzastopnim pozivom NEXT_CHAR može se pročitati čitava bejzik linija

7. LINE_ADDR CALL &196E

Potprogram LINE_ADDR služi za nalaženje adrese bejzik linije čiji linijski broj se ne ulazu nalazi u registru HL

Ne izlazu pra svega, treba proveriti indikator nule Z Ako je Z=1 onda je pronađene linije koju smo tražili i njena adresa je vrećene u HL Ukoliko je međutim Z=0 onde to znači da linija nije pronađena i e u HL će se nalaziti adresa prve sledeće linije u bejzik listi ili eko ni takve nema adresa odmah ize bejzik lisra (VARS)

U obe slučaja DE će na izlazu sadržati adresu linije koja prethodi onoj iz HL Ako takva ne postoji onda je DE=HL=PROG

Registar BC na izlazu sadrži ulazni linijski broj iz HL

8. NEXT_ONE CALL &1955

Ulazni parametar u ovaj potprogram je adresa proizvolne bejzik linije ili promenljive a treba je smestiti u registar HL Na izlazu ćemo u registru DE dobiti adresu sledeće linije odnosno promenljive praciznije adresu odmah ize linije (promenljive) koju smo naznećili na utazu Registar HL i delje će čuvati ulaznu adresu_e u BC ćemo dobiti dužinu zadate linije (promenljive) tako da je HL+BC=DE

Ako recimo neka bejzik linija počinje na &5D00, onde ćemo adresu sledaće dobiti sa

```
26    LD      HL, &5D00
      CALL    &19B8
```

Ne ovaj način znajući još de prva linija počinje na adresi PROG, možemo jednostavno pronaći adrese svih linija u programu što važi i ze promenljive koje inače počinju ne adresi VARS

Potprogram NEXT_ONE **ne ema se pozivati** ako HL stoji ne kraju bejzik bloka, tj. iza postednje promenljive (bajt &80)

9. ALPHA CALL &2C8D

Potprogram ALPHA proverave da li je neki karakter altabetski tj. de li predstavlje veliko ili malo slovo abecede Ulazi se sa kodom ispitivenog karaktera u akumulatoru

Svi registri uključujući i A su sačuvani e resultat testa nalazi se u indikatoru pranosa „C' Samo eko je on setovan redi se o slovu

PRINT ili ne donji deo ekrana gde se inače ispisuju poruke i raporti na štampač ili ne neko drugo odredište zavisi od toge koji je kanal trenutno otvoren Kao de informacije zalste teku nekakvim kenalima koji se po želji mogu otvarati i zatvarati A radi se u stvari o tome da su rutine koje obavljeju ulazno-izlazne poslova **univerzalne**, i mogu da opstuže bilo koji periferijski uređej Jednostavno u tim rutinama se vrši grananje u zavisnosti od toge koji periferijski uređej je aktivan A otvaranje kanala je prosto setovanje nekih bitova u memoriji, koji će biti tretirani u fazi granenja ulazno — izlazne rutine

Postoje četiri osnovne kanala, označene sa „K „S „R i „P', e povezuju računar redom sa tastaturom (i donjim delom ekrana) gornjim delom ekrana, radnim prostorom i štampačem

Jeden kanal obuhvata jednu ili više struje, kojima teku podaci Struje se označavaju brojevima a da bi se otvono neki kanal dovoljno je nevesti struju koja tom kanalu pripada.

Kanal „K obuhvata struje &00, &01, i &FD, kanal „S struje &02 i &FE kanal „R struju &FF, a kanal „P struju &03

O otvaranju kanala još ćemo detaljno govoriti e sada nas interesuje samo sedržaj bloka kanelskih mformacije počev od edrese CHANS Tu ukupno ima 20 bajtove po pet za sveki od četiri kanale Prva dva bejta sadrže adresu rutina koja tim kanalom šalje informacije na izlaz druge dve sadrže adresu rutine koja prime informactje sa ulaza, e peti bajt je oznaka samog kanala K „S „R ili „P'

Čitajući sadržaj kenalskih informacije videćemo de se izlazne rutine za kanale K , S i P nalazi ne adresi &09F4 dok je za kanal R to &0F81 Dalje ulazne rutine za kanal K' je &10AB dok ostali kanali nemaju definisan ulaz, pa tamo figuriše adresa &15C4 za izdevalje raporta „I Invelid I/O device

Pošto su kanelske informecije u RAM-u, **možemo** ih i sami menjati ubecujući umesto standardnih ulazno-izlaznih rutine neke svoje adrese. Ne žetost, to neće biti baš naročito jednostavno, jer spektrum neke adrese stalno obnevlja Jedino rešenje je de i mi svoje **edrese** stelno obnevljamo pa — ko duže izdrži

Iza bloka sa kanelima nalazi sa graničnik &80

SISTEMSKE PROMENLJIVE

Prostor između adrese 23552 i 23733 zove se blok sistemskih promenljivih, jer svaki od tih bajtova nosi neku važnu informaciju neophodnu ze uspešan red čiravog sisteme

Neke sistemske promentjive sastoje se od samo jednog bajta neke od dve bajta e neke i od više bajtova, u zavisnosti od toga kakvu informeciju čuvaju Na primer sistemska promentjiva zvana SUBPPC na edresi 23623 čuve redni broj bajzik neredba u linijli koja se upravo interpratira Za to je dovoljan jeden bejt. S druge strane, verijabla PPC sadrži sam tinijski broj linije koje se interpratira, e za to su potrebne dva bajta (23621 i 23622) Prvi bajt je uvek bajt menje težine

Čitenje sadržaje sistemskih verijabli vršićemo u bejzitu pomoću funkcije **PEEK**, e upisivanje novog sadržaja obavićemo neredbom **POKE** Neravno dobro treba proveriti da li upisujemo pravi broj na pravo mesto, jer posledice se inače ne mogu uvek predvideti

Spektrum neprestano u procesorskom registru IY drzi adresu 23610 i u odnosu ne nju odreduje relativni položaj ostelih sistemskih promenljivih Ne primer verijabla SUBPPC se nalazi na adresi tY+13

Dajemo pregled svih sistemskih varijebli sa kratkim opisom njihovog sadržaje Osim dekadne i heksadekadna edrese navedene je i relativne adresa u odnosu ne registar IY, a iza imene promenljive stoji i broj bajtova koje zauzima

23552	&5C00	IY-58	KSTATE	8	Koristi se prilikom očitavanje testature
23560	&5C08	IY-50	LAST_K	1	Posledni ukucani znak
23561	&5C09	IY-49	REPDEL	1	Vreme merano u pedesetim delovime sekunde posle koga počinje sutomet sko ponevljanje pritisnutog tastera
23562	&5C0A	IY-48	REPPER	1	Vreme merano u pedesetim delovime sekunde izmedu dva uzastopna ponavljanje pritisnutog tastera
23563	&5C0B	IY-47	DEFADD	2	Adresa prvog argumenta iza otvorane zagrade u neredbi **DEF FN**, ali samo eko je u toku izračunavanje funkcije ineče je sadržaj nula.
23565	&5C0D	IY-45	K_DATA	1	Kod za boju prilikom unošenja komendnih zneková dtrektno sa tastatura

23556	&5C0E	IY-44	TVDATA	2	Kodovi koji slede iza komandnih znakova prilikom njihovog ispisivanja
23568	&5C10	IY-42	STRMS	38	Tabela koja određuje kojoj struji odgovara koji kanal. Prva dva bajta u tabeli odnose se na struju &FD, druga dva na struju &FE, zatim na &FF, &00, &01 itd. U tim parovima bajtova upisan je **relativni pokazaj** zapisa odgovarajućeg kanala u bloku kanalskih informacija. Na primer, treći par bajtova sadrži broj &0008, a to znači da struji &FF odgovara kanal R-, jer njegov zapis počinja na **jedanaestom bajtu** u bloku kanalskih informacija.
23606	&5C36	IY-4	CHARS	2	Adresa umanjena za 256 na kojoj počinja tabela se ASCII znakovima. U ROM-u ova tabela počinja na &3D00, tako de CHARS sadrži &3C00. Možemo napraviti i svoju tabelu znakova. Svaki znak je predstavljen se osam bajtova odnosno osam linija iz kojih je sačinjeno znakovno polja na ekranu
23608	&5C38	IY-2	RASP	1	Trajanja zvuka upozorenja
23609	&5C39	IY-1	PIP	1	Trajanje tona koji se generiše pri pritisku bilo kog tastera
23610	&5C3A	IY+0	ERR_NR	1	Kôd za izdavanje raporta umanjen za jedinicu
23611	&5C3B	IY+1	FLAGS	1	Pojedini bitovi koriste se za indikaciju različitih stanja sistema prilikom interpretacija bejzika.
23612	&5C3C	IY+2	FVFLAG	1	Indikatori stanja pri radu sa TV
23613	&5C3D	IY+3	ERR_SP	2	Pozicija na mašinskom steku gde se nalazi adresa rutine za prijavljivanje grešaka.
23615	&5C3F	IY+5	LISTSP	2	Pozicija na mašinskom steku gde se nalazi adresa za povratak iz automatskog listanja
23617	&5C41	IY+7	MODE	1	Indikator trenutnog stanja kursora
23618	&5C42	IY+8	NEWPPC	2	Linija u bejzik programu na koju će biti obavljen skok (npr. **GO TO**).
23620	&5C44	IY+10	NSPPC	1	Redni broj naredbe u liniji na koju će biti obavljan skok. Ako je sadržaj &FF onda nema skoka
23621	&5C45	IY+11	PPC	2	Linijski broj naredbe koja se upravo izvršava
23623	&5C47	IY+13	SUBPPC	1	Redni broj naredbe u liniji koja se upravo izvršava
23624	&5C48	IY+14	BORDCR	1	Atribut za donji deo ekrana. Boja podloga je istovremeno i boja ivičnog dela ekrana.
23625	&5C49	IY+15	E_PPC	2	Broj bejzik linije u kojoj se nalazi programski kursor
23627	&5C4B	IY+17	VARS	2	Adresa bloka sa bejzik varijablama
23629	&5C4D	IY+19	DEST	2	Adresa bejzik varijabla kojoj se upravo dodeljuja vrednost.
23631	&5C4F	IY+21	CHANS	2	Adresa kanalskih informacija.
23633	&5C51	IY+23	CURCHL	2	Adresa trenutno važećeg kanala u bloku kanalskih informacija.
23635	&5C53	IY+25	PROG	2	Adresa bejzik programa
23637	&5C55	IY+27	NXTLIN	2	Adresa bejzik linija neposredno iza one koja se trenutno izvršava
23639	&5C57	IY+29	DATADD	2	Adresa iza poslednjeg pročitanog podatka u **DATA** listi
23641	&5C59	IY+31	E_LINE	2	Adresa editnog prostora
23643	&5C5B	IY+33	K_CUR	2	Adresa kursora
23645	&5C5D	IY+35	CH_ADD	2	Adresa karaktera (u bejzik liniji) koji je na redu da se interpretira
23647	&5C5F	IY+37	X_PTR	2	Adresa na kojoj je otkrivena greška u bejzik liniji
23649	&5C61	IY+39	WORKSP	2	Adresa radnog prostora

Treba imati u vidu da će se u povratak iz LINE_SCAN obaviti **same ako je aritmetika bila korektna.** U protivnom ćemo imati prakid sa raportom za grešku C Nonsanse in BASIC. Međutim, to se može izbeći ako se veštο iskoristi sistemska varijabla ERR_SP o čemu ćemo govoriti u okviru sledeće rutine

3 ERROR_1 — RST &08

Potprogram ERROR_1 ima zadatak da brzo i efikasno razreši sva problema nastale pri pojavi greška u interpretaciji bejzika. Možda se pitate šta tu uopšte ima problematično? Zar ne može računar, čim utvrdi grešku (npr. deljenje nulom) jednostavno da prekine rad i vrati se tamo odakle je dotična rutina bila pozvana? Svakako da bi mogao kada bi znao gde treba da se vrati. Problem je zapravo u tome što je mašinski stek, koji inače sadrži i adresu za povratak u međuvremenu verovatno zatrpan drugim adresama i podacima i pitanje je gde u toj gomili tražiti ono što nam treba. Zato se prosto pozova rutina ERROR_1, koja bukvalno poništi čitav mašinski stek i postavi ga na adresu sadržanu u sistemskoj varijabli ERR_SP. Ako nakim slučajem budete analizirali rutinu ERROR_1, videćete da se ona praktično samo sastoji iz:

```
LD      SP  (ERR_SP)
RET
```

Ostatak čine manje važni poslovi kao što su priprema pozicija za znak , ? u liniji se greškom prikupljanja koda za izdavanje raporta itd.

Pitanje koje se ipak nameće je šta se dešava posle naredbe RET? Gde se to obavlja povratak iz ERROR_1? Svakako na tamo odakle je usledio poziv sa RST &08 već na naku novu adresu koja se zatekla u vrhu mašinskog steka posle instrukcija **LD SP, (ERR_SP)**. Obično je to adresa za izdavanje raporta &1303 tako da svaka greška izaziva prakid programa.

Naredbu **RST &08** obavezno mora da prati još jedan bajt kojim se definiše tip greška. Vrednost tog bajta treba da je za jedan manja od stvarnog koda greška &FF za „0 OK" &00 za „1 NEXT without FOR" itd. Recimo ako u našem mašinskom programu na bilo kom mestu želimo prekid sa raportom „F invalid file name", postupaćemo kao u primeru 23

```
23    RST     &08
      DEFB    &0E
```

Međutim, cilj nam je da stavimo pod kontrolu i greška koja se mogu javiti dok se izvršava neki potprogram iz ROM-a. Želimo da grešku analiziramo i sami, pa da donesemo odluku da li treba ići na raport ili ne. Naćemo dakle da se posle ERROR_1 uvek odlazi na &1303 već na naku adresu u RAM-u, koju sami odaberemo.

Sva što treba uraditi je ubaciti željenu adresu na mašinski stek a u ERR_SP ubaciti pri tome dobijanu vrednost procesorskog registra SP. Razuma se prethodno ćemo upamtiti stari sadržaj ERR_SP, da bismo se kasnija uspešno vratili u bejzik (primer 24)

```
24    LD      HL  (ERR_SP)
      PUSH    HL
      LD      HL  naša adresa za greška
      PUSH    HL
      LD      (ERR_SP) SP
```

Posle ovoga, svaka greška će izazvati skok na našu adresu. Sistemska varijabla X_PTR sadržaće poziciju na kojoj je greška otkrivana tj. poslednja stanja varijabla CH_ADD. Sama greška tj. njen kôd umanjen za jedan nalazi će se u ERR_NR

4 REMOVE_FP — CALL &11A7

Potprogram REMOVE_FP uklanja iz bejzik linije sve binarna zapise numeričkih konstanti kao i odgovarajuće bajtove umetnute iza argumenate DEF FN funkcija. Na izlazu se zadaje početna adresa za pretraživanje a treba je smestiti u HL. Povratak iz potprograma će usledi čim se nađe na kraj linija (&0D, ENTER). Registri nisu očuvani. Izlaznih parametara nema a BC je na povratku uvek nula

5 GET_CHAR — RST &18

Ovaj potprogram koristi se za čitanje znakova iz kojih se sastoji bejzik linija. Ulaznih parametara nema a na izlazu ca akumulator sadržati karakter sa adrese CH_ADD. Registri su očuvani osim HL koji će i sam sadržati adresu CH_ADD

Međutim na radi se prosto o čitanju sa zadate adrese jer bi inače bila dovoljna i naredba **LD A,(HL).** Ovde na izlazu odbijamo samo **važeće bejzik znakove**, dakle ona se

očuvani a posle obavljanog štampanja broj biva izbrisan sa računskog staka PRINT_FP se na sma pozivati ako je stek prazan

Treba voditi računa i o jednom na baš beznačajnom _bagu koji je skrivan u ovoj rutini Naime ako je broj koji se štampa bio po modulu manji od jedinica tako da mu je celobrojni deo nula (ali sam broj je različit od nula) onda će na povratku iz PRINT_FP na računskom staku ostati zaboravljena jedna nula zamenjujući broj koji je otštampan! Upravo ja to razlog zašto spektrum na uma da izračuna recimo izraz oblika bilo koji niz +STR$ 0 5 Problem je u toma što funkcija STR$ i sema koristi PRINT_FP da bi _odštampala niz direktno u radni prostor prako kanala R Nula koja pri toma ostaje na steku rameti dalja računanja

1 OUT_LINE CALL &1855

Potprogram OUT_LINE obavlja stampanja bejzik linija izdvojena iz programa Tako na primer naredba LIST prosto poziva uzaslopno OUT_LINE dok se na izlistaju sve linije

Na ulazu se jadino zahteva početna adresa linija i traba ja staviti u registar HL a podrazumeva se kao i do sada da ja odgovarajući kanal otvoran Regitn na izlazu nisu sačuvani Primar 20 pokezuja kako sa moža odštampati prva bajzik linija iz programa

```
20      LD      HL (&5C53)
        CALL    &1855
```

Početak bajzik programa ja ovda uzet iz sistamske vanjable PROG Ako u programu na postoji ni jedna linija lil ja HL postavljan na kraj bajzik programa usledice povratak bez bilo kakvih problema

BEJZIK INTERPRETER

EDITOR CALL &0F2C

EDITOR je jedan od najvažnijih potprograma u spektrumovom ROM u Bilo šta da ukucavamo prako tastatura naredbu programsku liniju, ili INPUT liniju — sve će sa se to odvijati pod kontrolom ova rutina

EDITOR možamo pozivati i sami u okviru svog mašinskog programa kad god od korisnika očekujemo da unesa naki test Prathodno ćamo obriseti editni prostor (pozivom SET_MIN) pripremajuci ga tako za prihvatanja nova linija Osim toga obavazno se maka otvoriti kanal K a ako želimo K kursor moramo resetovati bit 3 vanjable FLAGS Inače ulaznin paramatara nema

```
21    CALL &165O      SET_MIN
      RES 3 (IY+1)    K-kursor
      SUB A           Struja &00
      CALL &1601      Kanal K — CHAN_OPEN
      CALL &0F2C      EDITOR
```

U dnu ekrana će se pojaviti kursor a korisnik može da pocne sa kucanjam bejzik linija Povratak iz editora ća usledi čim bude unet karaktar **ENTER** a ukucani tekst ostace u memoriji počev od adrese E_LINE što je u stvari početak editnog prostora Tamo možamo analizirati liniju znak po znak i proveriti njenu sintaksu

Izlaznih paramatara nema Registri nisu očuvani Posla svakog ukucanog znaka automatski se poziva rutina BEEPER tako da na izlazu iz editora nija sačuvan ni registar IX

2 LINE_SCAN CALL &1B17

Potprogram LINE_SCAN koristi se za sintaksnu proveru bejzik linije koja je prethodno bila unata u editni prostor recimo pomocu editora

```
22    CALL &0F2C      EDITOR
      CALL &1B17      LINE_SCAN
```

Ulaznih i izlaznih paramatara nama a registri nisu očuvani U toku sintaksne provare iza svake numerička konstanta kao i iza svakog argumanta u DEF FN naredbema biće umatnut bajt &0E pracan se još pet bajtova binernog zapisa

23651	&5C83	IY+41	STKBOT	2	Adresa računskog staka
23653	&5C85	IY+43	STKEND	2	Adresa iza računskog staka
23655	&5C87	IY+45	BREG	2	Sadržaj procesorskog B registra u trenutku poziva kalkulatora
23656	&5C88	IY+46	MEM	2	Adresa mamorijskog prostora koji sa koristi iz kalkulatora
23658	&5C8A	IY+48	FLAGS2	1	Indikatori različitih stanja.
23659	&5C8B	IY+49	DF_SZ	1	Ukupan broj redova iz kojih se sastoji donji deo ekrana.
23660	&5C8C	IY+50	S_TOP	2	Linijski broj programska linija od koja počinja automatsko listanja
23662	&5C8E	IY+52	OLDPPC	2	Linijsk broj na koji će skočiti naredbe **CONTINUE**
23664	&5C70	IY+54	OSPPC	1	Broj naredbe ne koju će skočiti **CONTINUE**
23665	&5C71	IY+55	FLAGX	1	Razni indikatori
23666	&5C72	IY+56	STRLEN	2	Dužina promanljiva kojoj se upravo dodeljuje vrednost (ako ja promenljiva znakovnog tipa)
23668	&5C74	IY+58	T_ADDR	2	Adresa sledećeg podatka u parametarskoj tabeli za vrama Interpretiranja nake bejzik naredbe Parametarska tabela počinja u ROM-u na adresi &1A7A i sadrži podatka o sintaksi svake bejzik naredbe
23670	&5C76	IY+60	SEED	2	Početna vrednost niza slučajnih brojeva kako ja to definisano narebom **RANDOMIZE.**
23672	&5C78	IY+62	FRAMES	3	Časovnik — brojač mašinskih prakida Svakih 20 ms brojač se uveća za jedinicu Prvi bajt ima najmanju težinu
23675	&5C7B	IY+65	UDG	2	Adresa grafičkin znakova
23677	&5C7D	IY+67	COORDS	2	‚x' i ‚y' koordinata poslednja tačke nacrtana na akranu
23679	&5C7F	IY+69	P_POSN	1	33 minus kolona na kojoj će štampač ispisati sledeći znak
23680	&5C80	IY+70	PR_CC	2	Adresa sledeće pozicija u bafaru za štampač koju će koristiti naredba LPRINT
23682	&5C82	IY+72	ECHO_E	2	33 minus kolona i 24 minus linija za poziciju u donjam delu ekrana koja odgovara kraju **INPUT** linija.
23684	&5C84	IY+74	DF_CC	2	Adresa sledeće PRINT pozicija u video mamoriji
23686	&5C86	IY+76	DFCCL	2	Kao DF_CC samo za donji deo akrana.
23688	&5C88	IY+78	S_POSN	2	33 minus kolona i 24 minus linija za sledeću PRINT poziciju
23690	&5C8A	IY+80	SPOSNL	2	Keo S_POSN samo za donji deo akrana
23692	&5C8C	IY+82	SCR_CT	1	Brojač vertikalnih pomaranja akrana (skrolovanja) Kada stigne do nula, ispisuja se poruka scroll?"
23693	&5C8D	IY+83	ATTR_P	1	Stalni atribut za glavni deo ekrana, onako kako je definisan naredbama za boju
23694	&5C8E	IY+84	MASK_P	1	Stalna maska postavljana naredbama **OVER, INVERSE** itd
23695	&5C8F	IY+85	ATTR_T	1	Privremani atribut, postavljan prilozima za boju u okviru jedna **PRINT** narebe
23696	&5C90	IY+86	MASK_T	1	Privremana „maska" za takuću PRINT naredbu
23697	&5C91	IY+87	P_FLAG	1	Indikaton
23698	&5C92	IY+88	MEMBOT	30	Šest mamorijskih lokacija od po pet bajtova koja koristi kalkulator
23728	&5CB0	IY+118		2	Na konsti se
23730	&5CB2	IY+120	RAMTOP	2	Poslednja adresa dostupna bejziku
23732	&5CB4	IY+122	P_RAMT	2	Poslednja adresa RAM-a

KOMUNIKACIJA SA KASETOFONOM

Kada govorimo o strukturi memorije i načinu na koji spektrum zapisuje razne informacije moramo nešto reći i o tome kako se organizuje snimak programa na magnetofonskoj traci. Naročito će nam ovo biti važno kada budemo direktno koristili mašinska rutine za rad sa kasetofonom o čemu će biti reči kasnije.

Bejzik naredba **SAVE** obavlja svakako jednu od najsloženijih operacija na spektrumu uopšte. Snimak sadržaj memorije na običnu kasetu i to tako da se sve kasnije bez ikakvih problema može uzeti natrag, zaista je vrhunac programerskog umeća.

Mi na žalost ne možemo ovde da se detaljno upoznamo sa procedurom snimanja već ćemo samo razjasniti neke najvažnije stvari.

Snimak se uvek sastoji iz dva zasebna dela. Bez obzira šta se šalje na kasetu prvo dolazi tzv. **zaglavlje**, a tek iza njega glavni blok podataka. Zaglavlje nosi informaciju o tome šta je snimljeno i sastoji se od 17 bajtova.

Prvi bajt zaglavlja (označimo ga sa B0) može imati vrednost **&00, &01, &02,** ili **&0.3,** zavisno od toga da li se snima **bejzik program, numerički niz, znakovni niz,** ili **prosto blok bajtova.**

Neredni deset bajtova (B1—B10) sadrže ime programa, odnosno bloka koji se snima. Ime mora imati makar jedan karakter a eko je kraće od deset ostatak se dopunjava praznim poljima.

Bajtovi B11 i B12 uvek sadrže dužinu bloka koji će biti snimljen, pri čemu B11 ima manju težinu.

Preostala četiri bajta zavise od tipa podataka koji se snimaju.

- **Bejzik program (tip &00)**

B13 i B14 sadrže linijski broj od koga će se program automatski startovati po učitavanju sa kasete (B13 ima manju težinu). Ako nema automatskog starta, onda B14 sadrži &80.

Bajtovi B15 i B16 sadrže dužinu samog programa, bez promenljivih.

- **Numerički i znakovni niz (tipovi &01 i &02)**

B14 sadrži ime niza, tačnije prvi bajt zapisa niza u prostoru bejzik promenljivih (redukovano ime i tip promenljive). Ostali bajtovi nisu značajni.

- **Blok bajtova (tip &03)**

B13 i B14 sadrže početnu adresu bloka u memoriji odakle će se bajtovi slati na kasetu. Preostala dva bajta ne nose nikakvu informaciju.

POTPROGRAMI IZ ROM-a

Sve što spektrum ume da radi zapisano je u njegovom ROM-u, u obliku mašinskih programa ukupne dužine 16 kilobajta. Tu se mogu naći uputstva za rad sa periferijskim uređajima, za prihvatanje, razumevanje i izvršenje bejzik naredbi i za izdavanje raporta o tome kako je izvršavanje proteklo.

Većina rutina je organizovana na principu potprograma, tako da se jednostavno mogu pozivati i koristiti kada god to zatreba. Na primer, „spektrum" poziva potprogram za štampanje znakova više od 20 puta u raznim delikama. Nikakav problem nije da i mi to uradimo u okviru svog mašinskog programa.

Pozivanje potprograma iz ROM-a se vrši naravno pomoću naredbe CALL (ili RST ako adresa to dopušta). Na primer, program za štampanje znakova na ekranu, nalazi se na adresi &0010 i pozivacamo ga sa **RST & 10**. Međutim, koji će se znak stampati? Odgovor na to pitanje je ovde prost: štampaće se onaj znak čiji se kod nalazi u akumulatoru mikroprocesora. Prema tome, ako hoćemo da štampamo recimo slovo „A" moramo uraditi **LD A, & 41** pre poziva rutine iz ROM-a. Na taj način pripremamo **ulazne parametre** neophodne za uspešan rad potprograma. Time međutim uopšte nisu rešeni svi problemi oko štampanja. Pre svega je ostalo najsasno kude će otići slovo „A" ne glavni deo ekrana (tj. tamo gde ineće radi bejzik naredbe **PRINT**), na donji deo ekrana (tamo gde se ispisuje raport) ili možda ćak na neki drugi periferijski uređaj (rutine za štampanje je ineće univerzalna).

Ovakvu nedoumicu treba rešiti još na samom početku našeg programe **stvaranjem odgovarajućeg kenala**. Za sada ćemo samo reći kako se to radi za glavni deo ekrana: u akumulator treba ubaciti &02 i onda pozvati rutinu &1601 (primer 3.)

```
3     LD      A, &02
      CALL    &1601
```

Čim se ove dva instrukcije izvrše nedalje će svako štampanje ići ne glavni deo ekrana, sve dok ne otvorimo neki drugi kanal.

Prema tome veliki deo posla oko poziva nekog potprograma iz ROM-a predstavljaju pripreme za taj poziv. Neke od pripreme su sasvim jasne (recimo zadavanje ulaznih parametara), ali neka su prilično skrivena. Dovoljno je nepravilni mali previd pa da nikad ne

Slika 20. Tabela sa porukama



Ulazni parametri u PO_MSG su adresa tabele, smeštena u DE i redni broj poruke u tabeli smešten u A. Redni brojevi počinju od nule.

Kao primer ćemo odštampati tekst „Nonsense in BASIC" koristeći tabelu sa raportima iz ROM-a, na adresi &1391. Redni broj izabranog raporta je &0C.

```
16    LD      A, &0C
      LD      DE, &1391
      CALL    &0C0A
```

Na izlazu iz potprograma očuvan je registar HL.

8. OUT_CODE **CALL &15EF**

Ovo je u stvari ista rutina kao i PRINT_A_1 samo što na početku akumulatoru dodaje &30 što je ASCII kôd za karakter „0". Posledica toga je da ćemo na ekranu dobiti cifru dekadnog sistema koja odgovara numeričkom sadržaju akumulatora ne ulazu (naravno samo ako je taj sedržaj izmedu 0 i 9). Leko se postiže de za vrednost akumulatora 0—15 dobijemo cifre heksadekadnog sistema (primer 17.)

```
17    CP      10
      JR      C,OD_OK
      ADD     A,7
OD_OK CALL    &15EF
```

Na izlazu iz OUT_COOE nije očuvan registar E.

9. OUT_NUM_1 **CALL &1A1B**

Potprogram OUT_NUM_1 služi za štampanje četvorocifranog celog broja koji se na ulazu nalazi u registru BC. Ispisivanje se obavlja bez vodećih blankove kao racimo u primeru.

```
18  LD BC, 7000
    CALL &1A1B
```

Na izlazu su sačuvani registri DE i HL.

Spektrum koristi ovaj potprogram da bi štampao linijski broj iza raporta za grešku. Preporučujemo vam da ipak ako već štampate cele brojeve, omogućite ispis sve do 65535 i uvedete izbor vodećih blankova. U tu svrhu traba koristiti potprogram &192A o kome nećemo inače posebno govoriti a koji štampa pojedinačne cifre dekadnog broja. Kompletan potprogram bi izgledao

```
19
DIGIT5  LD      BC, -10000
        CALL    &192A
        LD      BC, -1000
        CALL    &192A
        LD      BC, -100
        CALL    &192A
        LD      BC,-10
        CALL    &192A
        LD      A,L
        CALL    &15EF
        RET
```

Pre poziva DIGIT5 u HL trebe ubaciti broj koji se štampe a u registar E &FF za slučaj bez vodećih blankova, a &20 ako su vodeći blankovi potrebni. Registri ne izlazu neće biti očuvani.

10. PRINT_FP **CALL &2DE3**

Ova rutina se koristi za štampanje proizvoljnih brojeva. Neme ulaznih parametera, a broj koji se štampa mora biti na računskom steku. Izlaznih parametara neme: registri nisu

```
5  PLOT                    CALL &22DC
6  PLOT_BC                 CALL &22DF
```

PLOT je izvršne rutine odgovarajuće bejzik naredbe i obavlje crtanje tačka na ekranu Nema ulaznih parametara ali se podrazumeva da se koordinate _x" i y nalaze na računskom steku y na vrhu steka, a „x odmah ispod O upotrebi računskog steka govoričemo kada budemo objašnjavali rad spektrumovog kalkulatora

Ukoliko ulazak u rutinu obavimo na adresi PLOT_BC onda semi prethodno moramo da smestimo kodrdinate u BC (C=x B=y) Primer 13 dača tačku u centru ekrane

```
13    LD      BC, &5380
      CALL    &22DF
```

Izlaznih parametara iz potprograma neme e registri nisu očuvani

```
9  PRINT_A_1                RST &10
```

Potprogram PRINT_A_1 obavlje štampanje ASCII karaktera koji se na ulazu nelazi u akumulatoru Prihvataju se svi kodovi navedeni u Dodatku A ‚spektrumovog uputstva a u zavisnosti od kanele koji je trenutno otvoran znak ce se poslatt na odgovarajući izlaz Izlaznih parametara neme registri su sačuvani e ako se štampanje vrši na ekranu konstiće se ista pozicija koju bi inače uzela t naredbe **PRINT**

Kao primer ćemo odštampati slovo „A" u donjem delu ekrane (pr 14) Prethodno otvaramo kanel K e pre povratka pozivamo I WAIT_KEY jer bi u protivnom preko slove „A" odmah došao report

```
14    SUB     A
      CALL    &1601
      LD      A ‚A
      RST     &10
      CALL    &15D4
      RET
```

U bejziku se ovo isto izvodi se **PRINT # 0, „A"; PAUSE O**

Posle poziva PRINT_A_1 pozicija za štempanje se pomere odmah iza odštampanog znaka

Komendni znakovi se tretiraju na uobičajen način Recimo, bejzik naredbe **PRINT AT 7,13, PAPER 1, INK 5, „M"** ostvaruje se u mašinskom jeziku

```
15    LD      A  &16      AT 7 13,
      RST     &10
      LD      A,7
      RST     &10

      LD      A,13
      RST     &10
-----------------------------------
      LD      A, &11      PAPER 1
      RST     &10
      LD      A,1
      RST     &10
-----------------------------------
      LD      A  &10      INK 5,
      RST     &10
      LD      A,5
      RST     &10
-----------------------------------
      LD      A  &4D      M"
      RST     &10
```

```
7  PO_M                    CALL &0C0A
```

U slučajevima kada treba štempati neki tekst ili poruku koristi se potprogram PO_MSG Sva poruke moreju se smestiti u memoriju u vidu tabele Početak tabele se označava bajtom &80 III bilo kojim drugim čiji je bit 7 setovan Onda redom slede tekstovi poruka Poseidnji znak svake poruke takođe mora imati setovan bit 7 Ako se recimo poruka završava slovom „e odgovarajući bejt naše briti &61 već &E1

dočekamo povratak iz rutine koju smo brzopleto pozvali Jedne od takvih opasnih sitnica je sadržaj registra IY Ako on slučajno nije &5C3A (a to je adresa sistemske varijable ERR_NR) spektrum će se vrlo verovatno zbuniti On dosra često koristi registar IY jer u odnosu na njega određuje adrese svih sistemskih varijabli Najsigurnije je zato uopšte na koristiti IY u svom programu

Drugi deo posle oko rada sa ROM-om, jeste prikupljanje i tumačenje **izlaznih parametara**, tj onoge što nam potprogram vraće Neka rutine kao ona za štampanje nemaju nikakvih izlaznih parametara, ali racimo potprogram za traženje neke bejzik promenljiva će nam, razume se, vratiti adresu te promenljive, ili će javiti da nije našao ono što je tražio

I konačno treba znati još i koji od procesorskih ragistara će u toku potprograme biti sačuvani Ne retko ćemo sami morati da obavimo per PUSH instrukcije pre skoka kako bismo sačuvali ono što nam je važno Na primer

```
      PUSH    BC
      PUSH    HL
      CALL    NEKA ADRESA
      POP     HL
      POP     BC
```

Potprogram za štampanje &0010 ima tu lepu osobinu da čuva sve registre pa nikakve posebne manevre ne moramo preduzimati Ali takvih programa ja spak melo Sta više ima i takvih kod kojih ne vredi ni pamtiti neke stvari pra pozival Kakva je recimo korist od toga što upamtimo adresu bejzik linije 1000, ako pozovemo rutinu koja izbacuje lli ubacuje neka linije pomerajući rako i liniju 1000? Treba, dakle misiiti na sve

Izdvojili smo iz spektrumovog ROM-a semo najvažnije rutine koje se jednostavno mogu konstiti Za potpuno upoznavanje sa svim detaljima preporučujemo knjigu The complete Spectrum ROM Disassembly Iz koje smo inace i preuzeli sve nazive potprograma Nije loša ideja da ber provenra adrese jer svaka štamparska greška može da zeblokira već program i bez razloge vam pokvari raspoloženje

Potprogremi su podaljeni u grupe preme poslu koji obavljaju Iza imena svakog potprograma sledi naredbe za poziv se heksadekadnom adresom e zatim tekst sa objašnjenjima i primerima

Medu ulazno-izlaznim parametrima se često javljaju t indikatori nule i prenosa (zero flag i carry flag) koje ćemo označavati sa Z" i „C

Izuzev ako posebno ne neglasimo suprotno uvek se podrazumeva da stanje akumulatora ne izlazu **nije očuvano** Slično tome registar IX će gotovo uvek biti očuvan ‚Spektrum koristi ovaj registar samo izuzalno i to ćemo naglesiti kada bude trebalo

Neki potprogremi rade sasvim drugačije u fazi aintrakne provera bejzik naredbi nego u fazi **izvršenje** bejzik programa ‚Spektrum pravi razliku izmedu ova dva slučaja tako što konsti sistemsku vanjablu FLAGS eko je njen sedmi bit setovan onda je u toku izvršenje programe, a eko je resetovan onda se semo proveruva sinteksa Za korisnika je značajniji prvi slučaj, jer se svi mašinski programi obično izvršavaju u okviru bejzika — Konkretno za vreme izračunavanja funkcije **USR** Mi zato uglavnom nećemo ni pominjati detalje vezana za sintaksnu proveru

1 GLAVNA IZVRŠNA PETLJA

Čim uključimo spektrum on počinje de izvršava mašinski program iz ROM-a počev od adresa &0000 Tu ima niz poslova eko postavljanja sistemskiz varijabli i memorijske mepe e onda se ekran briše ispisuje poruka proizvodača i ulazi se u tzv glavnu izvršnu petlju Računar u njoj ostaje praktično sve dok ga ne isključimo

Nije teško zamisliti šta se dešave u okviru glavne petlje ‚spektrum prvo čeka de koriank nešto ukuca i pritisne ENTER Zatim proverave koliko tekst ima smisla po sintaksnim pravilima bejzika Konačno izvršava zadatu naredbu lli eko je bila uneta nova programska linija, dodaje je postojećem programu Bilo kako bilo, po ovavljenom poslu sa ponovo odlazi na početak glavne izvršne petlje i teko sve u krug

Glavna izvršne petlja nije nikakav potprogram Korisnik može samo de ostvari nepovraten skok u nju obično de bi izvršio resetovanje računare

```
1  START                   JP &0000
```

Skokom na adresu nula postiže se isti efekat kao keda se računar tek uključi Uništeva se sadržaj čirave mamorije e resetovanje je potpuno i efikasno.

2. NEW — JP &11B7

Skok na &11B7 ekvivalentan je izvršenju naredbe NEW iz bejzika. Procesor prelazi u interapt mod 1 a poništava se ceo bejzik prostor do RAMTOP a.

3. MAIN_EXEC — JP &12A2

To je početak glavne izvršne petlje. Obavlje se automatsko listanje bejzik programa kao da je pritisnut **ENTER** a u dnu ekrana ce se pojavi kursor.

4. MAIN_1 — JP &12A9

Ulazom na &12A9 se ed n zbevada auto tatsk stanje (na e je sve isto kao kod MAIN_EXEC.

2. PRISTUP MEMORIJI

U ROM u postoji niz potprograma pomoću kojih se vrlo prosto mogu ubaciveti ili izbacivati proizvoljni blokovi memorije. To sa pra svaga odnosi na prostor sa bejzik linijama i varijablama gde se statno nešto dodaje ili oduzima. Nije neravno problem izbrisati neku liniju iz programa i ubaciti novu. Problem je obračuneti novu dužinu programa i nove adrese svih blokove iza programa da bi se odgovarajuće sistemske varijable u skladu sa tim izmenile. Bilo bi zaista preterano očekivati od korisnika da sam o tome vodi računa. Oveko dovoljan je samo jeden CALL da bi se svi posiovi obavili.

1. RECLAIM_1 — &15E5

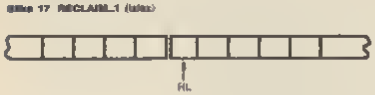
Potprogram RECLAIM_1 obavlja izbacivanje nekog memorijskog bloka. Na ulazu se u DE mora staviti adresa na kojoj počinje blok koji se izbacuje a u HL mora biti adresa iza tog bloka tj. prva adresa koju ne traba izbaciti (sl. 16.)

Slika 16. RECLAIM_1 (ulaz)



Na izlazu (sl. 17.) HL će sadržati novu adresu bloka koji se neizio iza izbačenog (to je zapravo ista adresa koja je bila u DE na ulazu).

Slika 17. RECLAIM_1 (izlaz)



U toku potprograma procesorski registri nisu očuvani, a na izlazu je BC uvek nula.

Uzecemo kao primer da iz bejzik programa treba izbeciti neku liniju koje počinje na adresi &5D00. Pri tom je, recimo, adresa sledeće linije &5DFF. Sva što treba uraditi je pripremiti HL i DE, i onda pozvati potprogram.

```
4   LO      OE &5D00
    LO      HL &5OFF
    CALL    &19E5
```

Na izlazu će linije iza izbečene biti pomerene na &5D00 i HL će sadržati tu adresu.

Recimo učitaćemo blok bajtova koji smo snimili u prethodnom primeru.

```
10  LO      A, &FF
    SCF
    LO      DE  30000
    LD      IX, 25000
    CALL    &0556
```

Na izlazu registri nisu očuvani, a indikator prenosa „C" je setovan ako je operacija obavljena uspešno do kraje. Međutim, ukoliko dobijemo C=0, to znači da je nastupila neka greške ili tip podataka na kaseti ne odgovara sadržaju akumulatore na ulazu, ili bajtove više name e očekivani broj je bio veći, ili konačno provera se kontrolnim bajtom nije uspela.

Pritisak ne taster BREAK_SPACE u toku čitanje kasete izaziva prekid. O BREAK-CONT repeats." Što sa tiče mašinskih interapte važi isto kao i za SA_BYTES na izlazu sa uvak obavije Ei.

4. KONTROLA EKRANA

1. CLS — CALL &0D6B

Ovaj potprogram je u stvari izvršna rutina bejzik naredba CLS i obavlja brisanje celog ekrana. Nikakvih ulaznih parametara neme, a registri ne izlazu nisu očuvani. Pozicije za **PRINT** se postavlja u gornji levi ugao ekrana, a za **PLOT** u donji levi ugao.

2. CLS_LOWER — CALL&0D6E

Rutina CLS_LOWER služi samo za brisanje donjeg dela ekrana. Koliko ce linija time biti obuhvaćeno — određeno je sadržajem sistemske varijable DF_SZ, koji možemo i sami menjati. Na primer ako treba izbrisati linije 17—21, dakle donjih pet na glavnom delu ekrana, onda, imajući u vidu još i dve linije u donjem delu ekrane, pre poziva CLS_LOWER treba u DF_SZ ubaciti broj sedam.

```
11  LD      (iY+49) &07
    CALL    &0D6E
```

Ulaznih i izlaznih parametara neme, a registri neće biti očuvani.

3. CL_SCROLL — CALL &0E00
CL_SC_ALL — CALL &0DFE

Pomoću potprograma CL_SCROLL vrši se vertikalno pomeranje ekrane za jedan rad naviše — skrolovanje. Jedini ulazni parametar je broj redova koji ce učestvovati u pomerenju i treba ga smestiti u registar B. Na primer ako u B ubacimo broj patneest, to znači da će 15 donjih redove (13 iz gornjeg i 2 iz donjeg dela ekrana) krenuti naviše. Linija 8 pri tome biva prebrisana (ne njeno mesto dotazi linije 9) a linije 0—7 ostaće nedirnute.

```
12  LD      B, &0F
    CALL    &0E00
```

Izlaznih parametara neme. Registri nisu očuvani.

Napominjemo da po obavljenom skrolovanju pozicija za štampanje **neće biti vraćena na red 21,** već to moramo sami uraditi, recimo sa **PRINT AT 21,0,** u bajziku, ili ne odgovarajući način u mašinskom jeziku, kao što će kasnije biti objašnjeno.

Ulez ne CL_SC_ALL konsti se za pomeranje **čitavog ekrana.** Nikakav ulazni parametar nije potreban, jer se automatski obavlja naredbe **LD B, &17**.

4. PIXEL_ADD — CALL &22AA

Svaka tačka na ekranu predstavljena je jednim bitom u video memoriji. Potprogram PIXEL_ADD proneiazi u memoriji odgovarajući bit za proizvoljnu tečku sa koordinatama „x" i „y". Na ulazu mora biti C=x a B=y. Ne izlazu će HL sadržati adresu bajta, a akumulator će odrediveti poziciju bita u okviru tog bajta: bit=7—A. Očuvan je registar DE.

Ukoliko je na ulazu „y" veće od 175, imaćemo grešku „B integer out of renge".

4. BEEPER — CALL &03B5
BEEP — CALL &00F8

Potprogram BEEPER koristi se za ganerisanje tona na „spektrumovom" ugrađenom zvučniku. Na ulazu se zadaje trajanja tona i njegova visina, ali ne baš tako jednostavno kao u bejziku.

Ako treba odavirati ton frekvencija „f" (u harcima) i trajanja „t" (u sekundama), onda sadržaj registara na ulazu mora biti

$$DE=f*t$$
$$HL=437500/f - 30.125$$

Naravno, rezultate treba zaokružiti.

Frekvencije tonova u temperovanoj lestvici dobijaju se po formuli

$$f=440*EXP((n-9)*l n 2/12),$$

gde je „n" ceo broj koji odgovara datom tonu, kao u slučaju bejzik naredbe BEEP. Recimo, da bismo iz mašinskog programa proizveli notu „C" (n=0, f=261.63 Hz) moramo postupiti kao u primeru 8.

```
8       LD      DE, 262
        LD      HL, 1642
        CALL    &03B5
```

Nema izlaznih parametara, registri nisu očuvani, a čak se koristi i register IX. Na izlazu su A i DE uvek nula. Pre povratka se uvek obavlja mašinska instrukcija EI, a u toku same rutine prekidi su onemogućeni. Ako vaš program zahteva isključena prekida, moraćete posle svakog poziva potprograma BEEPER ponovo da izvršite DI.

Drugi način ganerisanja zvuka je korišćenjem potprograma BEEP. To je izvršna rutina odgovarajuće bejzik naredbe, i radi malo sporije. Izlaz je isti kao i iz BEEPER, ali ulaznih parametara nema. Umesto toga trajanje note i njena visina moraju se naslaziti na računskom steku (visina note poslednja). O upotrebi steka tek ćemo govoriti.

5. SA_BYTES — CALL &04C2

Ova rutina obavlja snimanje blokova bajtova na kasetu. Nju naredba **SAVE** normalno poziva dva puta: prvo da bi snimila zaglavlje, a zatim i sam programski blok.

Na ulazu akumulator sadrži &00 ako se snima zaglavlje, a &FF ako se snima blok. U oba slučaja registar DE sadrži broj bajtova koji će se slati na kasetu, a IX njihovu početnu adresu. Izlaznih parametara nema. Registri nisu očuvani, a operacija sa u svakom trenutku može prekinuti pritiskom na taster BREAK-SPACE, što dovodi do raporta „D BREAK-CONT repeats".

Na kasetu se prvo snima jedan vodeći signal konstantnog tona koji traje oko 5 s za zaglavlje, a oko 2 s za programski blok. Prvi bajt koji se snima je oznaka tipa (zaglavlje ili blok), a zatim slede i sami bajtovi sa adrese IX. Iza njih se ubacuje još jedan kontrolni bajt dobijen logičkom operacijom **XOR** između svih snimljenih bajtova.

Kao primer ćemo snimiti na kasetu bajtove koji počinju na adresi 25000, se dužinom 30000. Snimamo samo blok — bez zaglavlja.

```
9       LD      A &FF
        LD      DE  30000
        LD      IX  25000
        CALL &04C2
```

Ovako snimljen program na može se učitati iz bejzika, jer nema zaglavlja. Još bolja zaštita je ako ceo blok snimimo pod oznakom zaglavlja (samo umesto **LD A**, &FF treba staviti **SUB A**). To će čak zbuniti i mnoga programa za presnimavanje!

U toku snimanja su onemogućeni mašinski prekidi, a na izlazu se obavlja EI.

6. LD_BYTES — CALL &0556

Potprogram LD_BYTES koristi se za učitavanje ili verifikaciju bajtova snimljenih na kaseti. Na ulazu akumulator sadrži &00 za zaglavlje, a &FF za blok. Indikator prenosa „C" je setovan za učitavanje, a resetovan za verifikaciju. Registar DE sadrži broj bajtova koji se očekuju sa trake, a u IX je početna adresa u memoriji gde traba uneti učitana bajtove, ili odakle ih treba verifikovati.

2. RECLAIM_2 — CALL &19E8

RECLAIM_2 je u stvari tati potprogram kao i RECLAIM_1, samo što se ulazni parametri razlikuju: sada HL treba da pokazuje na početak bloka koji se izbacuja, a u BC se mera smestiti dužina bloka (sl. 18.).

Slika 18. RECLAIM_2 (ulaz)



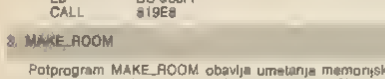
Izbacivanje bejzik linija se adrese &5D00 sada bi se izvelo kao u sledećem primeru.

```
5       LD      HL &5D00
        LD      BC &00FF
        CALL    &19E8
```

3. MAKE_ROOM — CALL &1655

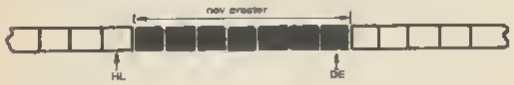
Potprogram MAKE_ROOM obavlja umetanja memorijskog bloka, pomarajući pri tome ostatak prostora udesno za potreban broj bajtova. Na ulazu HL sadrži adresu na kojoj će početi nov prostor, a u BC se nalazi veličine tog prostora.

Slika 19. MAKE_ROOM (ulaz)



Na izlazu HL pokazuje na bajt ispred novog prostora, a DE pokazuja na poslednji bajt novog prostora. Akumulator je očuvan a BC sadrži uvek nulu.

Slika 20. MAKE_ROOM (izlaz)



Kao primer umetnućemo prostor dužine &C0 na mesto izbačene linije se adrese &5D00 (primer 8.)

```
6  LD      HL &5D00
   LD      BC  &00C0
   CALL    &1655
```

Naravno, tek treba u taj prostor upisati tekst linije.

4. ONE_SPACE — CALL &1652

Kada traba napraviti mesta samo za jedan bajt, dovoljno je samo ubaciti adresu u HL i odmah pozvati ONE_SPACE. To je u stvari ista rutina kao i MAKE_ROOM, samo što sadrži i dodatnu naredbu **LD BC &0001**.

5. BC_SPACES RST &30

Kao sto smo ranije vec rekli izmedju adresa WORKSP i STKBOT u spektrumovoj memoriji nalazi se **radni prostor** za privremeno smeštanje podataka Pra upotrebe, radni proslor se mora otvoriti odnosno u njega se mora umetnuti potreban broj bajtova, potpuno slično kao u slučaju MAKE-ROOM

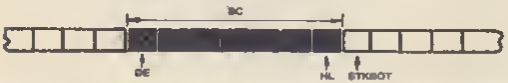
Otvaranje radnog **prostora** vrši se pozivom potprograma BC_SPACES Jedini ulazni parametar je dužina bloka koji želimo da umetnemo a treba da bude u registru BC Novi bajtovi će biti umetnuti u samom vrhu radnog prostora, odmah ispod STKBOT

Slika 21. BC_SPACES (ulaz)



Na izlazu DE pokazuje na prvi a HL na **poslednji** bajt novog prostora. Akumulator je očuvan a BC i dalje sadrži dužinu bloka

Slika 22. BC_SPACES (izlaz)



6 SET_MIN	CALL &16B0
SET_WORK	CALL &16BF
SET_STK	CALL &16C5

Potprogram SET_MIN koristi se za istovremeno brisanje editnog prostora, radnog prostora i računskog steka. Čitav prostor iza E_LINE će, tako, zauzeti minimalnu kontiguraciju

Ako treba obnoviti samo radni prostor i računski stak poziva se SET_WORK a za brisanje samog računskog staka koristi se SET_STK

Ulaznih parametara nema a u sva tri slučaja HL na izlazu sadrži novu vrednost STKEND Ostali registri uključujući i akumulator ostaju očuvani

7. TEST_ROOM CALL &1F05

Potprogram TEST_ROOM koristi se za provaru raspoloživog memorijskog prostora, dostupnog bajtiku Na ulazu BC sadrži broj bajtova se kojim se slobodna memorija testira Ako ima tolko praznih mesta, usledice povratak sa očuvanim registrima A i BC U protivnom obavice se skok na izdavanja raporta „4 Out of memory"

Rutina MAKE_ROOM i BC_SPACES sama pozivaju TEST_ROOM da bi proverila teren, tako da mi to na moramo činiti

3. PERIFERIJE

1. CHAN_OPEN CALL &1601

Spektrum obavlja ulazno-izlazna operacija sistemom kanala i struja o čemu smo već i govorili Da bi se precizirato kuda će otići izlazni podaci, ili odakle će se uzeti ulazni podaci mora se otvoriti odgovarajući kanal koji vodi do periferijskog uredjaja To je sve naravno samo sirkovit opis U suštini, otvaranje kanala se svodi na postavljanja sistemske varijabla CURCHL i setovanja nekih drugih indikatora (FLAGS i TVFLAG)

Otvaranje kanala obavlja se pozivom rutine CHAN_OPEN Ulazni parametar je numerička oznaka struje koja odgovara datom kanalu a treba ja smestiti u akumulator Kao primar ranije samo već imali otvaranje kanala _S končćanjem struja &02 Potpuno isti efakat bi se postigao i sa strujom &FE Sva ovo važi naravno samo ako na vršimo izmana u oblasti struja i kanala u RAM-u (edrese STRMS i CHANS)

Izlaznih parametara nema, a jedino je registar B očuvan

2 WAIT_KEY CALL &15D4

Potprogram WAIT_KEY, jednostevno čeka da korisnik pritisna neki taster t tak teda se vrši povratak To najčešće ima primenu u situacijama tipa , Press any key ", kada se prvo u dnu ekrana ispiše neka poruka i onda čeka akcija konsnika.

O ispisivanju poruka ćemo govoriti kasnija Sada je važno upamtiti da pra poziva WAIT_KEY obavezno treba setovati bit 5 u sistemskoj varijabli TVFLAG, jar će se inače u dnu ekrana pojaviti kopija bajtik iinija iz editnog prostora, što verovatno ne želimo

```
7     SET  6 (IY+2)
      CALL &15D4
```

Ulaznih parametara nema, svi registri ostaju očuvani, a akumulator na izlazu sadrži ASCii kôd ukucanog znaka (istovremano se taj kôd upisuja i u sistemsku varijablu LAST_K) Osim toga, zahvaljujući naredbi **SET 5, (IY+2)**, donji deo ekrana će biti automatski izbrisan čim taster bude pritisnut.

Podrazumeva se da ja pra poziva WAIT_KEY negda bio otvoren kanal „K (a sigurno će i biti ako već štampamo poruku u dnu ekrane) U protivnom ćemo imati grešku „J Invalid I/O device

3. KEY_SCAN Dec _ &028E

Ova potprogram obavlja očitavanja tastatura, ali za razliku od WAIT_KEY na čeka ništa već se vraća čim proveri svih 40 tipki Na izlazu ćemo tako dobiti podatak o toma da li ja trenutno neki taster bio pritisnut ili na

Kodovi koja vraća KEY_SCAN na odgovaraju ASCii znakovima, već **neposredno samim testerima** Na primer ako je pritisnuta tipka „A dobićemo na izlazu jedinstven broj bez obzira na to što se isti taster može tumačiti i kao „e ili kao „NEW , READ , itd Tek naknadno ako nam je potredno možamo dobijane kodova pratvarati u ASCii znakove, znajući stanje kursora Medutim sasvim ja dovojno u većini slučajava znati samo koji od 40 tastera je pritisnut da bi se preduzela odgovarajuća akcija.

Tipka su podaljana u osam grupa od po pet pri čemu spoljašnjt taster u svakoj grupi ima najveći kôd, a idući ka unutrašnjosti tastatura kôd opada se korakom osam Svi kodovi su dati u tabeli 1

Tabela 1

BREAK-SPACE	B	&20	&00
ENTER	H	&21	&01
P	Y	&22	&02
0	6	&23	&03
1	5	&24	&04
Q	T	&25	&05
A	G	&26	&06
CLAPS SHIFT	V	&27	&07

Pra poziva KEY_SCAN na moraju se vršiti nikakve pripreme, niti treba voditi računa o kanalima Ulaznih paramatara nema, a registri ne izlazu nisu očuvani Kodovi pritisnutih tastera biće smešteni u D i E a indikator nuše „Z biće resetoven ako ja pritisnuto više od dva tastera ili su pritisnuta dva a ni jedan od njih nija CAPS ili SYMBOL SHIFT Samo ako je pritisnut jedan tastar eventualno sa SHIFT-om, bica Z=1 Sadržaj registra DE ja sledeći

a) Ako nija pritisnut ni jedan taster, oba registra sadrža &FF Dovoljno ja testirati samo E

b) Ako ja pritisnut samo jedan taster makar i neki od SHIFT-ova, onda E sadrži njegov kôd, a D i dalje &FF

c) U slučaju dva pritisnute tastera jedan ida u E, a drugi u D Bilo koji SHIFT će obavezno ići u D a ako su oba tastera SHIFT-ovi onda CAPS ida u D



Spektrum

PRINTER

Svi koji su bili u prilici da prisustvuju dodeli nagrada na prošlom Geleksijinom Konkursu za najboljeg YU programere, mogli su da upoznaju jednog dečaka, čiji rad istine nije svrstan među četiri najbolje (pra svege zbog nepotpune dokumentacije) eli koji je zato izazvao veome povoljne reakcije za vreme demonstracije Reč je o Aronu Bošnjaku i njegovom programu Mastermajnd

Prisutni su slobodno možemo reći bili zadivljeni pojedinim efektnim rešenjima kojima je autor obogatio ovu popularnu igru učiniviši je tako po mnogo čemu različitom (i boljom!) od nekih drugih pa čak i komercijalnih rešenje

Aron Bošnjak je osvojio sedmo mesto e mi se iskreno nademo de njegov Mastermajnd ipak neće pesti u zaboravu De bismo vam dočarali bar nešto od onoga što ovaj program pruže odlučili smo da izdvojimo jednu kratku mešinsku rutinu koju je autor koristio za Ispisivenje uputetove pre početke same Igre Slove munjevito doleću sa desne strane ekrene i uz kerakterističen zvuk, zauzimaju svoje mesta, jedno ze drugim formirajuci reči Ideje je veoma duhovite i originalna e iz nje se dosta toge može i neučiti Zato smo odlučili da program izlistamo u asembleru (DEVPAC GENS) Morali smo de izvršimo i neke manje izmene u originelnoj rutini pre svega de bismo dobili nazeviaen potprogram koji bi se jednostavno pozivao iz bejzika Buduci de kao ulezni parameter more figurisati neki niz znekova koji će se štempati iskoristili smo metod DEF FN funkcije, o čemu emo već pisali u Računerima 3

Nerevno program možete koristiti i bez njegovog enelizirenja Jedino ga morete ukuceti liniju po liniju i zatim asemblireti (semi izebarite adresu na kojoj će se formireti mašinski kôd i unesite odgovarajuću ORG naredbu)

Ako je mašinski program unet recimo, na adresu 50000 onde u bejziku trebe ukucati liniju

10 DEF FN p(aŠ) – USR 50000

Ime funkcije i argumente naravno može bili i drugačije e adrese iza USR obavezno mora odgoverati prvom bajtu mešinskog programe

Time su eva pripreme obevljene Razume se mašinski kôd treba snimiti ne kasetu, ze kasniju upotrebu (dužina bloka je 117 bajtova)

Tekst koji se štempe pomoću rutine PRINTER pojevice se ne ekranu ne istom mestu gde bi ge inače smestile neredbe PRINT To zneči de se početne pozicije tekata može unepred definisati uobičajenim postupkom pomoću PRINT AT ili PRINT TAB Isto tako po završenom štampenju, PRINT pozicija će se nelaziti odmeh ize poslednjeg odštempenog zneka

Recimo da treba ispisati tekst , Galeksije Tada prosto treba izvršiti RANDOMIZE FN p(_Galeksija) Ili, eko je tekst koji se štempe amešten u verijubli h$ otkucaćemo RANDOMIZE FN p(h$)

Tekst duži od 32 zneka će se prenesiti u nov red Medutim, treba znati da će pokušej štempenje ize 21 rede dovesti do prekide se Izveštajem ,5 Out of screen e ne do poruke , acroll? Što bi se ineče moglo očekiveti Razlog je u tome što program PRINTER koristi zapravo neredbu PRINT AT u toku svog reda, a ove neredba, kao što korisnici ,epektuma dobro zneju odbije de redi sa 22 redom

Primetićete de se od svih znekove jedino prazne polja (blenkovi) ne ubacuju sa strene Oni se tretiraju melo drugačije i ne bi ih trabelo stevljeti na kraj niza koji se štempa jer će se ineče poremetiti PRINT pozicije za sledeće štampenje

Jovan Skuljan

```
;;;;;;;ram  PRINTEP
;Autor  Aron Bošnjek

START   LD      A,02            Biće otvoran kanel  S  (glevni deo ekrana)
        CALL    &1601           Otvaranje kanale
        LD      IX,(&5C0B)      DEFADD  edresa paremetre funkcije FN p
        LD      L,(IX+4)        Adresa niza koji se štampa
        LD      H,(IX+5)        Ide u registar HL
        LD      C,(IX+8)        Dužine niza koji se štampe
        LD      B,(IX+7)        Ide u registar BC
        LD      A,B             De li je dužine niza
        OR      C               jedneka null?
        RET     Z               Ako jeste — povratak u bejzik
        LD      A,24            Iz var jeble S—POSN (viši bajt) uzima se
        SUB     (IY+78)         linija ne kojoj će se štampati sledeći
        LD      E,A             znak Rezultat se smešte u E
        LD      A,33            Iz varijeble S—POSN (niži bajt) uzime es
        SUB     (IY+79)         kolone ne kojoj će es štampati sledeći
        LD      D,A             znak Pezultel se emešta u D
        SUB     32              De li je kolone 32?
        JR      NZ,CONT         Ako nije idi napred
LOOP3   LD      D,A             Broj kolon se postevlje na nulu
        INC     E               e pralazi se u sledeći red
CONT    DEC     D               Kolone koje prethodi tekućoj poziciji
        LD      (IY+71),D       smešta se u broječ
LOOP2   LD      D,31            Ubacivanje slova se vrši es desne etrane
LOOP1   LD      A,&18           To je kontrolni znak  AT
        RST     &10             PRINT AT
        LD      A,E             Broj reda es
        RST     &10             štempa iza ,AT  simbole
        LD      A,D             Broj kolone es
        RST     &10             ,štampe  iza broja reda
        LD      A,(HL)          Uzima se karakter iz niza koji es štampe
        CP      &20             De li je to blenk?
        JP      Z,INSN          Ako jeste — idi napred Neme usacivanja.
        RST     &10             Štempanje karektera
        CALL    TONE            Zvučni efekal
        LD      A,D             Testirenje kolone
        CP      31              De li je to poslednje kolena?
        JP      Z,NEXTLP        Ako jeste — idi napred
        LD      A,&20           Brisenje trega koj ostavlja za sobom elovo
        RST     &10             vrši es štempanjem blenka
        LD      A,&08           PPINT pozicije se odmeh vreća
        RST     &10             ze jedno mesto ulevo iza poslednjeg elova
NEXTLP  DEC     D               Sleva kliže ulevo
        LD      A,D             Testiranje kolone  de li je elovo stiglo
        CP      (IY+71)         na svoje mesto u tekstu?
        JP      NZ,LOOP1        Ako nije — idi nazad u petlju
INSN    INC     (IY+71)         Odštampano je još jedno slovo
        INC     HL              Pomeranje ne sledeći znak u nizu
        DEC     BC              Brojeć dužine niza es umanjuje es jedan
        LD      A,B             De li su odštempen svi
        OR      C               znakovi u nizu?
        RET     Z               Povratak eko jesu
        LD      A,(IY+71)       De li je popunjen
        SUB     31              čitav red ne ekranu?
        JP      NZ,LOOP2        Ako nije — idi nazad u petlju
        JP      LOOP3           Skok nazad  sa preleskom u nov red
TONE    PUSH    BC              Sečuvej broječ dužine
        PUSH    DE              Sečuvaj kolonu i liniju
        PUSH    HL              Sačuvej edresu zneka u nizu
        LD      A,32            Visina tone će biti edredena
        SUB     D               pomeću kolone na kojoj je
        RRA                     obavljeno štempenje
        LD      H,A             Dobijeni rezultet ide u HL
        LD      DE,&0001        Trajanje tone ide u DE
        CALL    &03B5           BEEP
        POP     HL              Obnavljenje registara
        POP     DE
        POP     BC
        RET                     Povratek
```

Galaksija

ŠIFRA

Zaštitom informacije bavili su se još stari Grci i Rimljani. Počelo je sa zaštitom (šifriranjem) pisanih poruka, ali danas to se proširilo ne samo na sve vrste informacije — govor, sliku, prenos podataka i sl.

Ovom problematikom su se pri svega bavili matematičari, ali su im u tome u dvadesetom veku pomogli mašinci sa mehaničkim šifrarskim mašinama, a zadnjih decenija i elektroničari.

Cilj ovog programa je da demonstrira mogucnost upotrebe računara - geleksija i za ovakvu namenu.

Pošto se program unese i startuje, po trebno je tekst koji se šifrira ili dešifrira uneti preme reseredu slove na testeru „geleksija". Šifrirane odnosno dešifrirane informacije se pojavljuje na ekranu televizore — monitore. Sveko slovo trebe držeti pritisnuto cca 1 do 2 sekunde (zbog petlje u liniji 199) što je učinjeno namerno jer prilikom skeniranje tastature vreme registrovanja pritiska na pojedine tastere nije isto pa je dolazilo do pojave dupliranja slova na ekranu.

Program je napravljen tako da se iste rutine koristi i za šifriranje i za dešifriranje. Sam ključ je moguće izmeniti tako što, na primer, sva slove pomerimo za jedno mesto nedole, a slovo „M" u red 101 teko de bude

101 IFK (1)P M
102 IFK (2)P N
103 IFK (3)P O
104 IFK (4)P P itd

Reči se resteviju pomoću blenke pri čemu se na ekranu pojevljuju tečke. Ispisani tekst na ekranu briše se pomoću horizontelne strelice koje pokazuje ulevo.

I na kraju jedna zaštićene poruka CBMQENI PVGNBPVZN ENPHANEN

Ivan Nador

```
50 L 500
101 IFK,(1)F,'N',
102 IFK, (2)F,'O'
103 IFK,(3)F,'P'
104 IFK (4)F,'R',
105 IFK,(5)F,'S',
106 IFK(6),'T',
107 IFK,(7)F,'U',
108 IFK,(8)F,'V',
110 IFK,,10)F,'W',
111 IFK,(11),' ',
112 IFK,(12)F,'Y',
113 IFK,(13)F,'Z',
114 IFK,(14)F,'A',
115 IFK,(15)F,'B',
116 IFK,(16)F,'C',
117 IFK,(17)F,'D',
118 IFK,(18)F,'E',
119 IFK (19)F,'F',
120 IFK,(20)F 'G',
121 IFK (21)F,'H',
122 IFK,(22)F 'I',
123 IFK,(23)F,'J',
124 IFK,(24)F,'K',
125 IFK (25)F,'L',
126 IFK,(26)F 'M',
131 IFK,(31)F,',',
132 IFK,(32)F,'4',
133 IFK,(33)F,'5',
134 IFK,(34)F,'0',
135 IFK,(35)F,'7',
136 IFK,(36)F,'0',
137 IFK,(37)F,'1',
138 IFK,(38)F,'2',
139 IFK,(39)F,'3',
140 IFK,(40)F,'J',
146 IFK,(41)F,'6',
188 IFK (29)C,5 0
199 F,X=0 TO 100,N:X
200 G,100
500 H,
510 F,AT 134,'*** SI RA ***'
520 F,AT 232,'NAIOK IVAN'
521 F,AT 3?9,' JUN 1985'
523 F,AT 394,'0, 3 1'
525 I X 0 TO 400:N,X
530 H,
540 F,'SIFRIRANJE ILI DESIFRIRA
NJE SE VRSI TEKU STO SE IFK I N
KULAVA GLEPAJUCI U TASTATURU PO
LAKO,'
550 F,'SLOVO PO SLOVO,SIFRIRANI
/DESIF RIRANI TEKST SE PISAV
JUJE NA EKRANU,'
640 F,AT 224,'TEKST se BRISE SIR
ELICOM'
655 F,AT 264,'NA LEVO '
660 F,
700 K,
```

Komodor 64

ARHIVA

Ako posedujete disk jedinicu sigurno imate bar jednu disketu popunjenu velikim brojem programa rednim verzijema i tekstovima koje retko koristite ili vam više neće trebati, ali biste jednostavno voleli da ih sačuvate. Ovim programom se tekve diskete mogu kopireti na kesetu i kasnije ako vam zatrebaju, ponovo vratiti nazad. U program je ugradene i opcije kopiranja diskete na disketu.

Kopiranje se vrši blok po blok i to cele diskete bez obzira da li je blok stvarno zauzet. Pošli smo od pretpostevke da će se ne ovaj način kopirati samo popunjene diskete pa bi izbegavanje slobodnih blokove nepotrebno kompilkovalo program.

Program je (za razliku od mnogih), mi ran u redu daje obeveštenje o toku (broj bloka koji se trenutno kopira) razlikuje programe ne treci od kopija diskete i može se prekinuti STOP tasterom. Proces kopiranje treke-disk ili obratno odvija se uz samo jednu intervenciju i traje oko 15 minute. Disk-disk zahteva 4 izmene diskete i traje oko 20 minute.

Keda unesete čitav program otkucajte RUN i računer će ispisiveti kontrolne sume koje trebe de uporedite sa datom listom. Ako se neka ne slaže odgovorite sa „N" i ne ekranu će biti izlistane grupa linije u kojoj je greška. Kada su sva kontrolna sume u redu, računer će sam izvesti proveru ukupnog zbire.

Seda program OBAVEZNO snimite ne disketu i u direktnom modu otkucajte

```
NEW
POKE25600,0
POKE44,100
NEW
```

I ponovo upišete program. Nekon

```
RUN 20
```

dobijete gotov oblik koji se snime i koristi uobičejenim procedureme.

```
1 REM ------------ A R H I V A ------------
2 REM
3 REM        C64 & 1541 & DATASSETTE
4 REM
5 REM AUTOR:  ZORAN ZIVOTIC  (85)
6 REM*-----------------------------------
7 REM PROVERA DATA LINIJA!
8 REM*-----------------------------------
9 RESTORE
10 FOR I=1000TO3556STEP41:S=0:FORJ=ITO3E
11 READB:S=S+B:NEXTJ:PRINT"S="I/4 25"=";S/
1E A$="D":INPUT  OK";A$
13 IFA$="D"GOTO15
14 PRINT"▲LIST":"- 1+3:POKE631 13
15 POKE632,13:POKE198 2 SYS42115
16 NEXT:PRINT"I  OK"
17 RESTORE:S=0:FORI=1TO2864:READB:S=S+B
18 NEXT:IFS<>219301THENPRINT"GRESKA!":END
19 PRINT"2  OK  CESTITAM":END
20 REM KREIRANJE PROGRAMA!
21 REM'-----------------------------------
22 RESTORE:FORI=2048TO4127:READB:POKEI,B
23 NEXTI
24 POKE44 9:POKE45,8:POKE46,18:CLR:LIST
100 DATA  0  48    8 193    7,158  50, 48
101 DATA 57  48  28  28  28  28  28  28
102 DATA 28  28, 48 173,173, 85, 82, 72
103 DATA 73  85  65 173,173  32, 98, 48
104 DATA 98  73  80  78  84, 73, 87,  0
105 DATA  8   6 185 142, 18,  3 169  14
106 DATA141  32 208,141, 33,208, 32,140
107 DATA 18 152,  0  32,158  14 169   8
108 DATA133 198, 32,228 255 240,251 201
109 DATA 48 144  38 201, 58,176,243  72
110 DATA 32 210,255, 32 143  10, 32 143
111 DATA 10 104,201  48,208   3, 76, 69
112 DATA  5,201  58 208   3  76,185,  0
113 DATA201  51,208  38  76  10,  9,201
114 DATA  3 208,207, 76,116 164  32,228
115 DATA255 240 251 208 100   0  32 117
116 DATA 10  48 174  10   3 154 174 238
117 DATA144 175  32,  13,  10,162   1  32
118 DATA156  14,  32  96,165 232,134 197
119 DATA208 132 199 162 255 232,169,  0
120 DATA  2 208 259 134 183, 32 143  10
121 DATA173   0   2 201  36,240, 24 162
122 DATA127  32,201 255 162  10, 32 156
123 DATA 14  32,204 255  56, 32, 74, 16
124 DATA 32 117  10,  56,  76,125,  0 185
125 DATA126,162,  0 100   0, 32,186 255
126 DATA 32 192,255,189,  0,133 144, 32
127 DATA146  10 162 126, 32,189 255, 32
128 DATA 52  10  32  52  10,173,141,  2
129 DATA208 251  32 195,  10, 32  52,  10
130 DATA 32  52,  10,  32,  52,  10,  72, 32
131 DATA 52  10 168 104,170,152,  32 205
132 DATA133   1 155  10  32  52  10  32
133 DATA210 255 200,240,  32 143,  10, 76
134 DATA225   0 162,  4,134,247,169,  1
135 DATA133,250  32,131,  10, 32,  85, 11
136 DATA 32  10,  11  32,131,  10, 32,  85
137 DATA 11  32, 30,  12,165,251,133,250
138 DATA198,247,240,238,169,  73,141   0
139 DATA  2,160,  0,141,  1,  2, 32, 13
140 DATA 10,  76  175,  0,102,  8, 32,156
141 DATA 14  32, 56,165,102,255 232,169
142 DATA  0,  2,157,  7  16,208,247,224
143 DATA  2,176   9  76,  57,  0,134 248
144 DATA 32,149,  10,  32, 56,250, 32 137
145 DATA 12,162   4 134 247,169   1 133
146 DATA250, 32 131,  10  32,  85,  11  32
147 DATA 15,  11,165-251,133 250,169,  0
148 DATA133 172 169  20,133,173 165,253
149 DATA133 174,165-254 133,175,169,  7
150 DATA133 107,169,  10 133,166,165 240
151 DATA133 183  32,146  13,130,247,200
152 DATA208,  24, 76,125   0,162,  4 134
153 DATA247 162   3,  32,  98,  11 169   1
154 DATA133,250  32,  16  14,165,  12 170
155 DATA160,141  53,  3,202 208,253 136
156 DATA208 250,206,  53,  3,208,245,169
157 DATA  0 133,172,160, 20,133,173, 32
158 DATA131  10,173,  62,  3,133,174,173
159 DATA 63   9,133,175, 32,100,  14 165
160 DATA144 240, 25 162   9  32,158,  14
161 DATA162   4, 32 156  14, 32 228,255
162 DATA201, 50,240   0,201,  70,208,245
163 DATA 24, 76,125   0, 32,  65  11,  32
164 DATA 43, 12 165 251,133,250,160,247
165 DATA200,169,  78,  32,  9 169,  0,133
166 DATA183 169 127,162,  0 160,  15  32
167 DATA186,255  32,192 255,  96,169,126
168 DATA162,  0,100.  2, 32,186 255 169
169 DATA  1,162 195,160,  14, 32,189,255
170 DATA 32,192,255,  96,  32 207,255,160
```

```
160 DATA144,200,  1, 96, 32 204 255  56
170 DATA 32, 74, 10, 32 117, 10, 56, 76
180 DATA125,  0,  0,162,127, 32,100,255
190 DATA100,  0, 32,207,255,153,  0,  2
200 DATA200,201, 13,208,245,169,  0,153
210 DATA  0,  2, 32,204,255, 40,170,  1
220 DATA 96, 32,140, 10, 32,143, 10,162
230 DATA 10, 32 156  14, 96, 32,204,255
240 DATA160,126, 32,165,255,165,127, 32
250 DATA195,255, 96,169,  0,133,253,165
260 DATA 20,133,254, 96,169,147, 44,169
270 DATA 13, 44,169,  5, 44,169,144, 44
280 DATA169, 46, 44,169, 32, 44,169,145
290 DATA 32,210,255, 96,165,251,162,  7
300 DATA 32,242, 10,165,250,162, 10, 32
310 DATA242, 10,162 127, 32,201,255,162
320 DATA 11, 32,156, 14, 32 204,255, 24
330 DATA 32, 74, 10,173,  0,  2,201, 58
340 DATA144,  5, 32,165, 10, 56, 96,162
350 DATA126, 32,100,255,100,  0,165,  1
360 DATA 41,254 133,  1, 32 207,255,145
370 DATA253,200 200 240  32 204 255,165
380 DATA  1,  5,  1,133,  1, 32,120  11
390 DATA 24, 96 100,165, 40,157,242, 15
400 DATA152,201, 10,144,  7,254,242, 15
410 DATA233, 10 176,165,  0, 40,157,243
420 DATA 15, 96,162,  2, 32, 96, 11,169
430 DATA 40,141,243, 15, 32,  13, 10, 32
440 DATA 30, 10,165,250,133 251, 32, 96
450 DATA 11,232,134,  2,162  0,134,252
460 DATA 32,184, 10,230,254,230,252,165
470 DATA  2,197,252,208 243,230,251,165
480 DATA251,100,245,208,225, 32,117, 10
490 DATA 96,162,135,240,  1,202,134,249, 96
500 DATA162, 96,201  10,144, 12,202,208
510 DATA201, 25,144,  5,202,201, 31,144
520 DATA  1,202, 96, 32,156, 14,169,  0
530 DATA133,100, 32,228,255,240 251,201
540 DATA  3,240,  1, 96, 24, 72,125-  0
550 DATA 32,146, 10,162,  7,189,242, 15
560 DATA 32,210 255,232,224, 12,208,245
570 DATA 32,143, 10, 32,150, 10, 32,225
580 DATA255,208,  4, 24, 76,125,  0, 96
590 DATA165,251,102,  7, 32,242, 10,165
600 DATA252,102  10, 32,242, 10,160,  0
610 DATA141, 52  3-165,251,201, 10,200
620 DATA 30,165,252,208, 26,165, 49,141
630 DATA 52,  3,141,243, 15,102,127, 32
640 DATA201 255,102,  11, 32,156, 14, 32
650 DATA254 255,169,  30,141,243, 15,102
660 DATA127, 32,201,255,102, 12, 32,156
670 DATA 14, 32 204,255,102,120, 32,201
680 DATA255,160,  0,165,  1 41 254,133
690 DATA  1,177-253, 32,210,255,173, 52
700 DATA  3,240,  6,102,143,208,  2,100
710 DATA255,200,200,232, 32,204,255,165
720 DATA  1,  5,  1,133,  1, 32,100, 11
730 DATA102,127, 32,201,255,102, 11, 32
740 DATA150, 14, 32,204,255, 24, 32, 74
750 DATA 10,173,  0,  2,201, 46, 24,240
760 DATA  4, 32,165, 10, 56, 96,102,  3
770 DATA 32, 00, 11 169, 50,141,243, 15
780 DATA 32, 13, 10, 32, 30  10,165,250
790 DATA133,251, 32,  00, 11,232,134,  2
800 DATA162,  0,134,252  32,152, 11,176
810 DATA 22,230,254,232,252,165,  2,187
820 DATA252,208,241,230,251,165,251,100
830 DATA249,208,223, 32,117, 10, 00,102
840 DATA  4, 32,156, 14, 32,228,255,201
850 DATA 00 240,222,201, 78,208,245, 32
860 DATA117, 10, 24, 76,125,  0,165,  1
870 DATA  3,  1,133,  1,165, 10,141, 32
880 DATA200,173, 17,200,  0, 10,141, 17
890 DATA200,165,  1,100,  1,  3, 32,133
900 DATA  1 132,102  05- 00,120,173, 17
910 DATA208, 41,239,141, 17,208 173, 32
920 DATA200-133,  10,100,  0,141, 32,200
930 DATA100,127,141,  0,220,165,  1, 41
940 DATA 31,100,  0,133,  1,132,102,102
950 DATA  0, 96,100,  0,169,  2, 32,210
960 DATA 12 162  - 176 102  3 200,244
970 DATA100,  5,100,171,200,230,152, 32
980 DATA210, 12,162,  7,136,208,247,232
990 DATA202, 05,133,109, 09,215,133,215
1000 DATA100,  0,133,103,  0,100,105,  1
1010 DATA 41,247, 32, 10, 13,162, 12,162
1020 DATA 12,034, 72, 44,  1 220, 48,  3
1030 DATA 76,130, 13,173, 32,200, 73,  1
1040 DATA141, 32 200,104,  9,  0, 32, 10
1050 DATA 13,162, 14,100,103,200,213, 00
1060 DATA202 200,253,144,  5,102, 11,202
1070 DATA200,253,133,  1  06, 32,146, 10
1080 DATA 32, 23,240, 32,140, 12, 00,169
1090 DATA  0,132,215,169,  7,141,  0,221
1100 DATA162  1, 32,100, 13, 30,103,165
1110 DATA100,201,  2,200,245,100,  9, 32
1120 DATA 61, 13,201,  2,240,249,100,100
1130 DATA200,232, 32,  01, 13,136,200,246
1140 DATA 00-100,  0,133,103, 32,100, 13
1150 DATA 30,100,234,234,234,100 163,200
1160 DATA244,185,100, 00,173, 32,200, 73
1170 DATA  1,141, 32,200,100, 16, 44,  1
1180 DATA220, 40,  3, 76,138, 13, 44, 13
1190 DATA220,240,243,173, 13,221,142,  7
1200 DATA221, 72,100, 25,141, 15,221,104
1210 DATA 74, 74, 00, 32,110, 12, 24, 72
1220 DATA125,  0,162,  7,134,171, 32, 56
1230 DATA240, 32,140, 12,105,  1, 41,254
1240 DATA133,  1, 32,100  12,100,  3, 24
1250 DATA101,247,133,105,202,202, 32,210
1260 DATA 12,102,  0,105,172,  0, 30,210
1270 DATA 12,102,  6,200,102,  5,234 200
1280 DATA242,100,  0,102,  4,177,107,100
1290 DATA103,144,  3,100, 32,202, 32,210
1300 DATA 12,102  5,200,132,107,200,237
1310 DATA100,  3,133,171, 32,100, 12,152
1320 DATA 32,210, 12,132,215,102,  7,234
1330 DATA177,172  32,210  12,102,  3,230
1340 DATA172,200,  4,230 173,202,202,105
1350 DATA178,107,174,105,173,225,175,144
1360 DATA231,234,105,215, 32,210, 12,162
1370 DATA  7 136,000,240, 32,110, 12, 00
1380 DATA 32, 25, 13, 32, 39, 13,201,  0
1390 DATA240,245 133,171, 32, 01, 13,145
1400 DATA178,200,152,102,200,240,100,  0
1410 DATA141,136,  3, 32,110, 12,102,  5
1420 DATA 32 150, 14, 00,100,  0,229,171
1430 DATA170,165,  0, 32 205,105, 32,132
1440 DATA 10,162, 13, 32,156, 14, 32,143
1450 DATA 10,165,171,201,  3,144, 12, 50
1460 DATA233,  3 107,247,208,  1, 96,102
1470 DATA  7,208,  2,162,  0, 32,156, 14
1480 DATA 50, 78 125,  0, 32, 29, 13,165
1490 DATA  1, 41,254 133,  1, 32, 30, 13
1500 DATA 32, 01, 13,145 172,132,144,132
1510 DATA144,132,144, 60-215,133,215,230
1520 DATA172,200,  2,230 173,165,172,157
1530 DATA174,165,173,225,175,144,225, 32
1540 DATA 01, 13, 32,110, 12,165,105, 60
1550 DATA215 133,144, 00,100 167, 14,100
1560 DATA100,101, 14  32  30 171, 55  14
1570 DATA 15, 15, 15, 15, 15, 15, 15, 15
1580 DATA 15,  2, 15, 15,  3,100, 04, 03
1590 DATA 97,120,150,150,103,213,223,  0
1600 DATA242,255, 65, 35, 13, 13,  5, 32
1610 DATA 32, 65, 32,  02, 32, 72, 32, 73
1620 DATA 32, 00, 32, 05, 32, 13,144, 32
1630 DATA102,192,192,132,102,192,192,102
1640 DATA192,192 192,132,192,102  13,  5
1650 DATA 32, 40, 40, 32,144, 60, 73, 03
1660 DATA 75, 45, 04, 02, 05, 75, 05, 13
1670 DATA  5, 32, 50, 40, 32,144, 04, 02
1680 DATA 05, 75, 05, 45, 00, 73, 03, 75
1690 DATA 13,  5, 32, 51, 40, 32,144, 60
1700 DATA 73, 03, 75, 45  00, 73, 03, 75
1710 DATA 13,  5, 32, 52, 40, 32,144, 00
1720 DATA 75, 03, 13, 32, 03, 40, 32,100
1730 DATA132,152,102,132,102,102,102,152
1740 DATA102,102  13,145, 32,  0, 17,144
1750 DATA 50, 73, 03  75, 50,  5,  0,  5
1760 DATA 52, 52, 32, 02  02, 32, 00, 70
1770 DATA 03, 04, 05, 00, 73, 32, 70, 02
1780 DATA 73, 71  73, 70, 05, 70, 32, 00
1790 DATA 73, 03, 75, 00, 04, 05, 40, 13
1800 DATA  0,144  02, 52, 32, 02, 02, 32
1810 DATA 00, 75, 03, 04, 05, 00, 73, 32
1820 DATA 00, 02, 03, 00, 70, 05, 32, 00
1830 DATA 73, 03, 75, 03, 04, 05, 40, 13
1840 DATA  0,144, 70, 05, 03, 04, 05, 00
1850 DATA 73, 03, 75, 03, 04, 05, 40, 13
1860 DATA 32, 25, 13, 32, 39, 13,201,  0
1870 DATA240,245 133,171, 32, 01, 13,145
1880 DATA178,200,152,102,200,240,100,  0
1890 DATA141,136,  3, 32,110, 12,102,  5
1900 DATA 32 150, 14, 00,100,  0,229,171
1910 DATA170,165,  0, 32 205,105, 32,132
1920 DATA 10,162, 13, 32,156, 14, 32,143
1930 DATA 10,165,171,201,  3,144, 12, 50
1940 DATA233,  3 107,247,208,  1, 96,102
1950 DATA  7,208,  2,162,  0, 32,156, 14
1960 DATA 50, 78 125,  0, 32, 29, 13,165
1970 DATA  1, 41,254 133,  1, 32, 30, 13
1980 DATA 32, 01, 13,145 172,132,144,132
1990 DATA144,132,144, 60-215,133,215,230
2000 DATA172,200,  2,230 173,165,172,157
2010 DATA174,165,173,225,175,144,225, 32
2020 DATA 01, 13, 32,110, 12,165,105, 60
2030 DATA215 133,144, 00,100 167, 14,100
2040 DATA100,101, 14  32  30 171, 55  14
2050 DATA 15, 15, 15, 15, 15, 15, 15, 15
2060 DATA 15,  2, 15, 15,  3,100, 04, 03
2070 DATA 97,120,150,150,103,213,223,  0
2080 DATA242,255, 65, 35, 13, 13,  5, 32
2090 DATA 32, 65, 32,  02, 32, 72, 32, 73
2100 DATA 32, 00, 32, 05, 32, 13,144, 32
2110 DATA102,192,192,132,102,192,192,102
2120 DATA192,192 192,132,192,102  13,  5
2130 DATA 32, 40, 40, 32,144, 60, 73, 03
2140 DATA 75, 45, 04, 02, 05, 75, 05, 13
2150 DATA  5, 32, 50, 40, 32,144, 04, 02
2160 DATA 05, 75, 05, 45, 00, 73, 03, 75
2170 DATA 13,  5, 32, 51, 40, 32,144, 60
2180 DATA 73, 03, 75, 45  00, 73, 03, 75
2190 DATA 13,  5, 32, 52, 40, 32,144, 00
2200 DATA 75, 03, 13, 32, 03, 40, 32,100
2210 DATA132,152,102,132,102,102,102,152
2220 DATA102,102  13,145, 32,  0, 17,144
2230 DATA 50, 73, 03  75, 50,  5,  0,  5
2240 DATA 52, 52, 32, 02  02, 32, 00, 70
2250 DATA 03, 04, 05, 00, 73, 32, 70, 02
2260 DATA 73, 71  73, 70, 05, 70, 32, 00
2270 DATA 73, 03, 75, 00, 04, 05, 40, 13
2280 DATA  0,144  02, 52, 32, 02, 02, 32
2290 DATA 00, 75, 03, 04, 05, 00, 73, 32
2300 DATA 00, 02, 03, 00, 70, 05, 32, 00
2310 DATA 73, 03, 75, 03, 04, 05, 40, 13
2320 DATA  0,144, 70, 05, 03, 04, 05, 00
2330 DATA 73, 03, 75, 03, 04, 05, 40, 13
2340 DATA  5, 70,144, 41, 13,  0,144, 70
2350 DATA 70, 00, 70, 00, 50,  5-  0, 13
2360 DATA144, 70, 73, 74, 00, 32, 75, 73
2370 DATA 00, 73, 74, 05, 32, 00, 73, 03
2380 DATA 75, 00, 04, 00, 40, 13,  0, 13
2390 DATA144, 00, 73, 71, 02, 00, 03, 05
2400 DATA 70, 32, 02, 05, 00, 72, 73, 32
2410 DATA 00, 02, 75, 74, 32  00, 70, 73
2420 DATA 75  00, 45, 13,  0,-13,144, 70
2430 DATA 65, 30, 73, 00, 50,  5,  0, 13
2440 DATA144, 04  05, 00, 05, 32, 02, 00
2450 DATA 65, 00, 32, 05, 02, 02, 70, 02
2460 DATA 13,  0, 05  50, 50, 50, 32, 48
2470 DATA 32, 51, 53, 32, 45  34,  0, 00
2480 DATA 45, 00, 50, 50, 32, 40,  0,  0
2490 DATA  0,  0,  0,  0,  0,  0,  0,  0
2500 DATA  0,  0,  0,  0,  0,  0,  0,  0
2510 DATA  0,  0,  0,  0,  0,  0,  0,  0
READY
```

KONTROLNE SUME:

```
S 0 = 2110      S 32 = 2779
S 1 = 2563      S 33 = 2645
S 2 = 4088      S 34 = 5164
S 3 = 3848      S 35 = 3234
S 4 = 3757      S 36 = 3376
S 5 = 3452      S 37 = 3259
S 6 = 3440      S 38 = 4580
S 7 = 2407      S 39 = 2868
S 8 = 3312      S 40 = 3883
S 9 = 2841      S 41 = 3389
S 10 = 3337     S 42 = 4539
S 11 = 3115     S 43 = 3607
S 12 = 4827     S 44 = 3071
S 13 = 3425     S 45 = 3688
S 14 = 3718     S 46 = 4172
S 15 = 3207     S 47 = 4803
S 16 = 4281     S 48 = 3547
S 17 = 4213     S 49 = 3781
S 18 = 3320     S 50 = 3058
S 19 = 3248     S 51 = 3151
S 20 = 3752     S 52 = 4441
S 21 = 4002     S 53 = 2384
S 22 = 3157     S 54 = 3052
S 23 = 3817     S 55 = 2676
S 24 = 3193     S 56 = 1907
S 25 = 5102     S 57 = 3178
S 26 = 4235     S 58 = 2113
S 27 = 3584     S 59 = 2146
S 28 = 3890     S 60 = 1993
S 29 = 3731     S 61 = 2214
S 30 = 4105     S 62 = 1890
S 31 = 4672     S 63 = 1057
S 32 = 2779     S 64 = 313
```

FUNKCIJSKI TASTERI

Tastetura „komodora" je ukrašena sa 8 funkcijskih tastera eli oni uglavnom zvrje neiskorišćeni Ovim programom dobijaju funkciju ze koju su i namenjeni — programiranje sadržeje koji će biti ispisan jednim pritiskom ne odgovarajući taster

Uvedene u dve nova naredbe

1)DEF n,STR$ kojom se odreduje sadr zaj tastera (n = broj F tastera STR$ = sadržaj)

2) kojom se dobija pregled sadržaje

U naredbi)DEF oba parametra mogu biti i promenjive Dužine stringe je ogreničena ne 127 znakova, o čemu sâm program vodi računa — eko je duži sledi poruka STRING TOO LONG

Definisanje i korišćenje tsatere moguće je u direktnom i programskom modu Ta ster f1 je redi ilustracije programiren za promenu boje Često mogu biti od koristi i sledeći oblici

)DEF 3, „OPEN4 4 CMD4 LIST '+CHR-$(13) +, PRINT & 4 CLOSE4 + CHR$(13)— listanje programe ne printeru

)DEF 5 „LOAD¯ + CHR$(34) + „$ + SHR$(34) + „,8" + CHR$(13) — za upis direktorija diska

Pri ispisu isprogramiranog sadržeja pojedinih tastera karekteri 13 i 141 ce biti ispisani kao znak?

Program koristi izmenu interapt vektore Ako se u toku rede upotrebi prekid sa STOP+RESTORE F tasteri će posle toge biti neaktivni Ipak njihov sadržej nije izgubljen i dovoljno je tražiti samo pregled se j da bi se opet aktivirali

Pre stertovenje program obavezno snimite jer se u njemu izvodi naredba NEW Ako dobijete poruku SYNTAX ERROR negde ste pograšili pri unošenju DATA linije

Zoran Životic

```
1 REM*------------ P - TASTERI ------------
5 BEM*                  -- 8 64 --
3 REM
4 REM     Autori Zoran Zivotić (85)
5 REM
6 REM  Nare4bel J
7 REM                  )DEF n,str$  Gin(8
8 REM*-------------------------------------
10 BEGTORE1=PEEK(PEEK(+48)+256)TO48818
11 BEADA12POKE1,A:M=M+A:NEXT
18 IPM<>35876?8THENPOKE38 255:BY344888
13 BYB48152
14 POKE631,?8:POKE632,68:POKE633,87
15 POKE634 :31POKE198,41:LIST-8
16 "
100 DATA189 E5,162 18C 141,8,9,142,0,9,1
E8,169 208,162 18E 141,20,3 14E E1
101 DATA3 60 00,32 115,8,E81,83,248,0,32
 1E1,8,78 231,187,3E 10,152 3E 115
102 DATA0 201,150,248 84,169,178 182,193
,168 0 133 E51,154 E5E,132,253,169
103 DATA154,188,183,32,38 171 185,E53,17
0-E4 185,40,52,210,E55,188,44,32
104 DATA218 295,169,34,38,E18,255,168,18
```

```
0,195,E48,E4 178,188,0,177 E51,7E
105 DATA41,1E7 201 13 208 4,184,169 63 7
2 184,32 210,255,208 287,208,233
108 DATA165 34 32,210 255 165 1E0,24,181
 251,153 251 144,8,E38 25E E30 253
107 DATA185 259,E01 0 288 177 76,174,187
3E 115 0-3E,198,183,138 E40,4 224
108 DATA5 144,5 18E,14 76,55 184,134,239
32 253 174 3E,158 173 32,183,18E
109 DATA201 128,144 5 182,E9,78,55 184,9
2 165 E55 166,E59 E62,185,183,157
110 DATA188 153,3E 1E0,169,188 0 177,187
 145 251,E88,188,185 E80,247 76,174
111 DATA167 ,3E,E34 255 185,204 208 41,18
8,285,E88 37,188 E8,133,285 184 211
112 DATA78,287 174 135 E 177,E89 178,17,
238,287 133 288,32,38 234,177,243
113 DATA141,135 E-174 134,2,185 E88 73 1
E8 3E,28 234,165,1,41,18,E48 10 188
114 DATA8 132,152 185 1 8,3E E88 8 165 1
82 288 8 165 1,41,31,31,133,1,173 188
115 DATA183 288 E3 3E 135 E34 159 128,24
0,47 185 118 E 201 133,144,48 E81
118 DATA141 178 38 58 E33,133,E81-4,144
5,E33 3 18,144 3 18 185,1,178 188
117 DATA158 183,E48 15 142 188,183,134,E
58,169 8 141 188 193,188 188,32 1E8
118 DATA185,173 13 E28 184 188,184,178,1
84,64 166 188,E6A 18,178,E41 17E
119 DATA188,183 177 251 157 119,2,238,15
8 238 185,198 288,174,188,183,152
120 DATA2E1--158 133,144 E18 188 8,141,18
8,183,E48,E11 185,178,E81 188,183
1E1 DATA251 134-E5E,158 253,E48,13,185,1
28,84,181 251,155,E51-144,E48,E38
1E2 DATA252 178,E38 5E 13,85,68,88,78,8,
32,8,8 8 8 8 8,8 8 87,81,48,32
123 DATA58 68,75,75 88,83,81,58-58 48,44
,87,58-88 75 75,88 53-51,58,58-48
1E4 DATA44 87 141,145,E8,E8,8
```


RAČUNAR GALAKSIJA ROM 8K, RAM 6K (u delovima)

Komplet sadrži sve potrabna komponente uključujući memorij u epirom procesor i sve druge elektronske komponenite testeturu štampanu ploču kutiju za računar i sprevljeć spekovano u plastičnoj embalaži Štampena ploča je tako uradena de omogućuje lako proširenje RAM i ROM memorije Uz sve to još i detaljno uputstvo za semogradnju knjige BEJZIK ZA GALAKSIJU (N Petezenovic) ROM—2 (Đ Ristenović) dve kasete sa programima (SUPER ŠAH i DEMOKASETA sa 14 programa)

Idesine prilika da do svog prvog računara dodete sopstvanim trudom po pristupačnoj ceni Komplet je namenjen i školama u kojima se na sekcijı ili u redovnoj nastavi (odredenih usmerenje) mogu sklopati računari i tako sticati znenja o njihovim funkcionelnim delovima

Za računar GALAKSIJa obezbeden je tehnički servis i stalna stručna pomoć

NARUDŽBENICA

Ovim neopozivo naručujem _____ komade računara galaksija u delovima sa pratećom opremom po ceni od 60 000 00 d-nara po komadu

NARUČILAC

MESTO

ULICA I BR

POTPIS

Ispunjene narudžbenice slati na adresu ZAVOD ZA UDŽBENIKE I NASTAVNA SREDSTVA — BEOGRAD Obilićev Venac 5/I tel 011/637 915 I 638-405

ZAVOD ZA UDŽBENIKE I NASTAVNA SREDSTVA — Beograd

Obilićev Venac 5 I tel 011 636-971 i 638-405

NARUDŽBENICA

Neopozivo naručujem

a) KAČUKASI

1 **GALAKSIJA 8—6** cena 7E 500 00 kom
2. **GALAKSIJA 8—6** u delovima (komplet deleva — uputstvo — knjige Bejzik za GALAKSIJU i ROM-2 i dve kasete sa programima) cena 60 000 00 kom
3 Pojedinačno delove sa računar GALAKSIJA

E) LITERATURA

1 **BEJZIK ZA GALAKSIJU** — (prel dl N Perezanović) cena 700,00 kom ___
2 **KUĆNI KOMPJUTERI** — Algoritmi i programi (bejzik spektrum) — mi N Mladenovic mi V Petrović Đ Grašvić cena 280 00 kom ___
3 **ZX SPECTRUM — PROGRAMIRANJE U BASICU** mi N Marković i D Davidović cena 750 00 kom ___
4 **PROGRAMIRANJE ZA POČETNIKE (Knjiga I)** — P Creckal I (prevod se engleskog) cena 750 00 kom ___
5 **PROGRAMIRANJE ZA POČETNIKE (Knjiga 2)** — P C sokal (prevod se engleskog) cena 750 00 kom

c) PROGRAMI

Za GALAKSIJU

1 **DEMOKASETA** (14 programa) cena 1200 00 kom ___
2 **SUPER ŠAH** cena 1200 00 kom ___
3 **MATEMATIKA** (4 programa Sedenje Oduzimenje Množenje i Deljenje) i TEST IZ OPŠTE TEHNIČKE KULTURE (4 ag em su namenjeni proveri stečenih znanje učenika osnovnih škola) cena 1200 00 kom ___
4 **MATEMATIKA — MNOŽENJE** (E program namenjenih učenju eve računske operacije za učen ke mlađih raz edeosnovne škole) cena 1200 00 kom ___

Za KOMODORE 64

1 **MATEMATIKA — MNOŽENJE** (Program namenjen učenju eve računske operacije za učen ke mlađih raz edeosnovne škole) cena 1200 00 kom ___

Za SPEKTRUM

1 **MATEMATIKA — MNOŽENJE** (Program namenjen učenju eve računske operacije za učenike mlađih razreda osnovne škole) cena 1200 00 kom ___
2 **MATEMATIKA** (Program namenjen proveri stečenih znanja učenike osnovnih škola u vladanju računom računar m operacijama) cena 1200 00 kom ___
3 **PIZIKA** (E **programa** Kretanje meteora u gasovima Karnoev ciklus Stepena iskost Kinematika Fotoelekat Program su namenjeni demonstraciji pri kom p edavanje u srednj m i osnovnim školama) cena 1200 00 kom ___
4 **ASTRONOMIJA** (4 **programa** Hale eva kometa Sunce i Mesec Planete Jupiterovi sateliti Prog am omogućuju da se pratnju posmatranja nebeskih tela i uvid u prošle i buduće položaje pomenutih tela) cena 1200 00 kom ___
5 **ENGLESKI JEZIK** (E programa namenjenih učenju engleskog jezika za početnike proveru stečenih znanja Visok stepen interaktivnosti pruža velike mogućnost za brzo i lako učenje) cena 1200 00 kom ___
6. **MATEMATIKA** (2 **programa** Manje više Aoskva Ogigu i Inventivne ke šćene grafike najmlađ m kori sn cima računare data je mogućnost da vežbaju logiku tablicas r elementarna računska radnja) cena 1200 00 kom ___
7 **MATEMATIKA — SEĆENJE** (Kaseta sadrži E cel na za učenje najlaše računska radnja a namenjena je najm ađ m učen cima osnovne škole) cena 1200 00 kom ___
8 **TEHNIČKO OBRAZOVANJE — ANALITIC** (Program je namenjen za izvođenje složenih aktivnosti (sekcija) u školama Klubove vazduhoplovnog modelarstva Za dale u sena vrednosti dobija se pratećim modelsav ena) cena 1200,00 kom ___

NARUČILAC

MESTO

ULICA BR

POTPIS

Ispunjena narudžbenice slati na adresu Zavoda

KAKO PRILAGODITI VIEW (II)

Vlasnici računare BBC koji su pažljivo pročitali „Računare 3" i pokušali da prilagode svoj tekst procesor VIEW jugoslovenskim slovima su se svakako našli u velikom čudu. Tehničkom greškom uz tekst je bio objavljen program koji u samom tekstu nije bio ni pomenut ni objašnjen a osnovnog (dobro objašnjenog) programa nije ni bilo!

Greške poput ove se događaju kada postoje stari i novi tekst a zatim se od tehničkog urednika minut pred zaključivanje liste traži da uzme novi tekst jeden program iz priloga starog teksta i drugi program iz priloga novog teksta. Tada nerevno sve ispadne neopeko — objavi se stari tekst i novi program!

Ne sluti je deo program za definisanje karaktera na ekranu većeg BBC B koji, osim toga, odebira potrebne opcije ze rad tekst procesora. Po njegovom startovanju (sa SHIFT BREAK ako imate disk ili *RUN 'Boot ako se služite kasetofonom) tasterima na kojima je nacrtane otvorene srednje zagrade, funta, zatvorene srednje zagrade i „backslash" biveju dodeljene naša slova č ć ž i š respektivno

Da bi se neša slove prenele ne papir potreben je nerevno, i printer drejver U Računarime 3 je objavljen drejver za Epson RX80 (radi i ne FX80 premda za njega može da se napiše daleko bolji program) koji će odgovarati i vlasnicime mnogih štampače koji su kompatibilni se Epsonom. Ovaj drajver osim pisenje neših slova omogućava podvlačenje tekste koje se koristi kao Highlight 2 prema uputstvu za korišćenje VIEW-a

Drejver za štampač Seikosha GP A koji smo najavili ne objavljujemo jer nam se čini da nije interesenten ze širi krug vlesnike BBC-ja. Ukoliko se neki od njih interesuje za program od Redakcije može da traži bespletan listing

Dejan Ristanovic

```
850 REM  Prva DATA lista daje string
860 REM  koji treba da se izvrsi pa
870 REM  inicijalizacija VIEW-a.
880 REM  String se zavrsava tackam
890 REM  a , oznacava RETURN.
900
910 DATA n.a.W.:.P.r.l.n.t.@.r." ",E.p
.s.o.n." "
920
930 FOR I=1 TO 90:READ A:?(&30FF+I)=A
940 NEXT I
950
960 *SAVE "!Boot" 3000 3160
970 END
980
990 REM  Serija DATA tablica koja
1000 REM  sadrza definicije karaktere
1010 REM  tj. bajtava koji se direktne
1020 REM  preslaljuju ruticni OSWRCH.
1030
1040DATA 23,96,12,60,102,96,96,102,60,0
1050DATA 23,123,108,60,102,96,96,102,60
,0
1060DATA 23,95,12,24,60,102,96,102,60,0
1070DATA 23,91,36,24,60,102,96,102,60,0
1080DATA 23,92,36,24,60,96,60,6,124,0
1090DATA 23,93,36,24,124,12,24,48,126,0
1100DATA 23,124,40,60,96,60,6,102,60,0
1110DATA 23,125,40,124,12,24,48,96,124,
0
```



```
10 REM
20 REM
30 REM                V I E W
40 REM
50 REM     Definicije karaktera  2.2
60 REM
70 REM              11.06.1984.
80 REM
90 REM
100 OSWRCH=&FFE3
110 OSBYTE=&FFF4
120 CLI=&FFF7
130 FOR I=0 TO 3 STEP 3
140 P%=&3000
150 [ OPT I
160 LDA #&9   \ rezervise strane &6 I
170 STA &349  \ &C za definicije.
180
190 LDX #0
200 LDY #&6   \ definise sve 8 slova
210           \ tj. salje 96 bajtav.
220 .CYCLE
230 LDA &3100,X
240 JSR OSWRCH
250 INX
260 DEY
270 BNE CYCLE
280
290 LDA #144  \ izvrsava *TV 255,1
300 LDX #255
310 LDY #1
320
330 JSR OSBYTE
340 LDA #22   \ izvrsava VDU 22,6
350 JSR OSWRCH\ tj. MODE 6
360 LDA #6
370 JSR OSWRCH
380
390 LDA #15   \ *FX 15 0 brise bafera
400 LDX #0
410 JSR OSBYTE
420
430 LDX #0
440 .EXNEW    \ stavlja NEW CTRL M
450           \ i PRINTER ... u
460           \ bafer tastature.
470 LDA &30A8,X
480 CMP @ASC "."
490 BEQ CONT
500 TAY
510 TXA
520 PHA
530 LDA #138
540 LDX #0
550 JSR OSBYTE
560 PLA
570 TAX
580 INX
590 BNE EXNEW
600
610 .CONT
620 LDA #202  \ bira malu slova.
630 LDX #49
640 LDY #0
650 JSR OSBYTE
660 JSR OSBYTE
670
680 LDX #(TEXT MOD 256)
690 LDY #(TEXT DIV 256)
700 JSR CLI   \ izvrsava *WORD
710
720 RTS       \ startuje VIEW
730 .TEXT EQUS "WORD"
740 EQU8 13
750 EQUS "(C) Dejan Ristanovic 1984"
760 EQU8 "."
770 I
780 NEXT I
790
800 I=&30A0
810 REPEAT READ X$:X=ASC X$
820 IF X$=',' THEN X=13
830 ?I=X
840 I=I+1:UNTIL X$="."
850
```

BBC B LATINICA NA „KANONU"

Ovaj program služi za definisanje YU slova na štampačime „Kanon" i „kogo" vezan je ze tekst „kanon" protiv „epsone" koji ob jevljujemo ne streneme 8 i 9

```
10 REM
20 REM
30 REM  Definicije YU stand. slova
40 REM
50 REM
60 REM          Verzija 0.1
70 REM
80 REM
90 REM     Dejan Ristanovic 1985.
100 REM
110 REM
120 ADP=&A30
130 AD=ADP
140 ?AD=ASC "S":REM definicije standardnih slova
150 AD=AD+1
160 OSWRCH=&FFEE
170 OSBYTE=&FFF4
180 FOR I=0 TO 3 STEP 3
190 P%=AD
200 [OPT I
210 LDA #3

220 LDX #10
230 JSR OSBYTE
240 LDX #TABL MOD 256
250 .SALJI
260 LDA ADP AND &FF00,X
270 CMP #&FF
280 BEQ RET
290 JSR OSWRCH
300 INX
310 BNE SALJI  \  uvek JMP
320 .RET RTS
330 .TABL
340 ]:NEXT I
350 AD=P%
360 REPEAT
370 READ A:?AD=A
380 AD=AD+1
390 UNTIL A=255
400 OPT I 2

410 OSCLI "SAVE Stand "+STR$~(ADP)+" "+STR$~(AD+1)+" "+STR$~(ADP)
420 END
430 DATA 27,64:REM Inicijalizacija
440 DATA 27,79:REM Iskljucivanje preskakanja perforacije
450 DATA 27,69:REM Prasirivanje svega znakova koji se ispisuju
460 DATA 27,58,0,0,0:REM Interni set slova u RAM
470 DATA 27,37,1,0:REM Aktiviranje PC8 karaktera
480 DATA 27,38,0,123,126:REM Redefinisu se ASCII kodovi 123-126
490 DATA 139,24,36,66,128,66,128,66,0,36,0,0:REM CACAK
500 DATA 139,36,82,0,210,0,210,0,82,12,0,0:REM SABAC
510 DATA 139,0,66,4,194,0,210,32,66,0,0,0:REM ZAPAC
520 DATA 139,0,0,28,34,0,98,128,34,0,0,0:REM celija
530 DATA 27,38,0,91,93:REM Redefinisu se ASCII kodovi 91-93
540 DATA 139,0,0,28,162,64,34,60,162,0,0,0:REM cacak
550 DATA 139,0,16,170,64,42,64,170,4,0,0,0:REM sabac
560 DATA 139,0,34,132,98,8,98,144,34,0,0,0:REM zabac
570 DATA 27,38,0,96,96:REM Redefinise se ASCII kod 96
580 DATA 139,24,36,66,0,66,128,66,0,36,0,0:REM CELIJA
590 DATA 255
```

Test sa zadrškom: prema svecu i drajv

Mikrodrajv nakon godinu dana

Fizički mikrodrajv se predstavlja kao mala crna naprava dimenzije 9 × 8 × 5 cm. Nije ni naročito lepa ni zgodno oblikovana. Pogled u unutrašnjost ne otkriva neku veliku mudrost: jedan motor, jedna magnetna glava, malo elektronike i dva konektora. Kertridž koji se ubacuje sa prednje strane nije ništa drugo do minijaturna kaseta (4.3 - 3 × 0.5 cm) koja sadrži 12 metara dugačku traku širine svega 2 milimetra. Cela stvar radi na principu beskonačne petlje: a Sinklar je to rašio tako što su na sredini kasetice (kertridža) nalazi jedan mali ločkić na koji je namotana traka. Sa jedne strane se traka odmotava i prolazi ispred magnetne glave, a sa druge strane se namotava.

Ipak ne puca

Sva to izgleda krajnje daleksto i krajnje — napouzdano! Ako se još zna da se traka kreće brzinom od skoro dva metra u sekundi, onda čoveku padne mrak (čitaj: PAPER 0) na oči. Ipak, koliko je poznato autoru ovog teksta, još niko u Beogradu nije imao problema sa mikrodrajvom! Razni slučajevi da traka puca verovatno su sa dašavali samo sa prvim primercima koji su se pojavili na tržištu. Varovali u Sinklars III na sa mikrodrajvom traba postupati jako pažljivo — isto učstalom vazi i za najskuplja disk jedinica. Što se kapacitata tiče, on vrlo čudno varira između 89 i 91 KB. Ako sa to nekome učini malo moguce je povazati do osam mikrodrajva u rednu vezu i time dobiti maksimalan kapacitet od 720 K, što nije nimalo zanemarljivo.

Hopa cupa preko rupa

Brzina prenosa podataka između spaktruma i mikrodrajva je viša nego izvanredna: 14 KB u sekundi. S obzirom da je ta brzina retko kad veća i kod najboljih disk jedinica, mogło bi se zaključiti da je mikrodrajv vrlo brza jedinica apoljne memorija. Na žalost nije tako. Da bi se neki podatak učitao traba najpre stici do njega. Prosečno vreme pristupa iznosi 3.5 sekunda a maksimalno 7 — to je vreme za koja se obrne cela traka. Iz toga proizlazi da sa program dugačak recimo 20 K učitava u proseku za 5 sekundi a u najgoram slučaju za 7. Ako je kasetica prazna ili vrlo malo popunjana to je tačno. U protivnom atvari se komplikuju. Naime, kod snimanja programa računar traži najveću praznu rupu na kaseticl i tu snima program. Ako ta rupa nije bila dovoljno velika traži sledeću i tu nastavlja snimanja. Jasno što je program iseckaniji to će njegovo učitavanja trajati duža — često mnogo duža nego onih maksimalnih 7 sekundi. U najgorem slucaju učitavanja 20 K dugačkog programa ume da potraja i dobrih 15 sekundi. Tranutak u poredenju sa kasatofonom. Ali pravi disk.

Naredba SAVE sa još duža izvršava nago LOAD. Računar mora da obrna celu traku da bi se uverio da za novi program ima dovoljno masta. Zatim dolazi već opisno hopa cupa preko rupa i na kraju poruka OK. Zavisno od slučaja to traja 10 do 20 sekundi. Ali to nije sva. Ako program se istim imenom već postoji na kertridžu (prosto račeno stara verzija programa) potrabno ga je prvo obrisati pomoću ERASE naredbe. To traja koliko i učitavanja. Najzad, kod vrlo važnih programa nije na odmat verifikacija jar se to ne vrši automatski prilikom snimanja. Da sa vrstimo na naš 20 K dugačak program — jedna operacija tipa ERASE/SAVE/VERIFY u praksi ratko kad traje manja od 30 sekundi.

Sintaksni teror

Ipak nija sve tako crno. Ma koliko rad sa mikrodrajvom izgladao spor, on je ipak nauporedivo brži nago sa kasetofonom. Što je najvažnija, sve se odvija potpuno automatski, nama onog pramotavanja kaseta, tražanja programa, podešavanja jačine, TAPE LOADING ERROR poruka i slično, što rad sa kasatofonom čini tolliko naprijatnim. Istina, dašava se da program sa mikrodrajva neće da se učita kako traba, ali to je vrlo raiko. U celini uzav, radi se o vrlo sigurnoj jedinici apoljne mamorija.

Postoji još jedna prednost mikrodrajva u odnosu na kasetofon: to je rad se datotekama. Operativni sistam podržava do 12 datoteka istovramano. Pri toma, svaki podatak može biti proizvoljne dužina, ali pristup, na žalost, može biti samo sekvancialan. Postoja još naka ograničenja. Recimo na poslop naredba koja bi prapoznala kraj datoteka, tako da se softvar koji kontroliše mikrodrajv teško može mariti sa nakim boljim DOS-om.

Ono što je stvarno katastrofalno kod mikrodrajva nije ni spor pristup, ni mali kapacitat ni sakvencialna datoteke. Katastrofalna je sintaksa naredbi. Da bi se učitao naki program sa mikrodrajva, potrabno je otkucati ni više ni manja nego

LOAD *"M", 1, „ime"

Ona jedinica, na primar, označava broj mikrodrajva. Pošto na „spaktrum" može biti priključano do osem mikrodrajva, navođanja log broja je sasvim opravdano. Ali, zaboga, zašto čovak atalno mora da kuca tu prokletu jedinicu ako ima samo jedan mikrodrajv? Ili ona zvazdica koja kod nekih naredbi mora da sa navede s kod drugih ne. Još gori primar je naredba CAT (katalog) koja umesto da prikaža kompletne podatke o svim programima koji sa nalaze na kertridžu izbacuja samo njihova imana. Izgleda kao da su naredbe radane tako da okana zagnžanog korlanika od mikrodrajva.

Dos, tos ili samo — SOS

Da bi se mikrodrajv povezao sa spaktrumom, potreban je interfejs 1 koji sa priključuje na port za proširanja sa zadnje strane računara. Pri toma, „spaktrum" dobija vrlo fini nagib koji olakšava kucanje. Ono sto, madutim, nija nimalo fino je dužina kabla koji povezuja interfejs i mikrodrajv svega osam centimatara. Zahvaljujući tome mikrodrajv sa može nalazili samo sa lave strane spaktruma — tamo gde je konaktor s na i sa desna. Ser Klajv u svojoj akscantričnosti varovalno laksa ubacuje kasatica lavom rukom.

Pogled u interfejs 1 otkriva jedan nerzbezni ULA čip, dva standardna TTL kola, jedan ROM — nazovimo ga ROM 2, par tranzistora i to bi otprilika bilo sva.

U 8K koliko sadrži taj ROM 2, upisan je kompletan program za rad sa mikrodrajvom koji bi uslovno mogao da se nazove TOS — tape operating system. Na žalost, baš u koncepciji tog TOS-a leža glavni već spomanuti, nedostaci mikrodrajva. Vaclna problama potiče još iz glavnog ROM-a, gde su kao u šopskoj salati katastrofalno izmešani operativni sistam i bajzik interpretar. Što je najgora, iako se za vrama pisanja glavnog ROM-a i ta kako dobro znalo za planirani mikrodrajv, ništa nije uradano da mu sa pripremi taren. Kao razultat ROM 2 uvodi naredbe za rad sa mikrodrajvom na nake gotovo nevarovatne načina, jar, naravno, nigde u glavnom ROM-u nija pradvidan ni jedan jedini link a praksa pomašanog operativnog sistama i bejzik interpretara je nestavljana. Rezultat je konfuzija u kojoj se čak i najiskusniji programari teško snalaze.

Sračom prosačnog korisnika na zanima mnogo šta i kako radi ROM. Za njaga je samo važno to da se svi programi izvršava ju baz problama. Na žalost ni to nija uvak slučaj. Praksa pokazuje da dobar deo uslužnih programa i većina igara neće da radi kako trabe kada se na „spaktrum" priključi interfajs 1. Razlog leži u toma što je početak bajzika pomeran sa 23755 na 23613 a mnoga atvari u području sistamskih promanljivih i kanalskih informacija se manjaju. Uz to mikrodrajv može da zahteva i do nakoliko kilobajta radna memorija, zavisno od toga šta se sa njim radi. Bilo kako bilo, vrlo verovatno ćata dobar deo svog postojaćag softvara morati da prilagodite novim uslovima.

Disk na grani, mikrodrajv u ruci

Ostaja još da sa razmotri odnos mogućnosti/cana. Tranutno sa u Englaskoj komplet ZX INTERACE 1 + MICRODRIVE + 9 praznih kasatica prodaja za oko 65 funti. Što je vrlo prihvatljivo. Tim pra ako se zna da razni kasatofoni tipa „data recordar" koji ni izdaleka nlau ravni mikrodrajvu dostižu cenu i do 40 funti. Ako dalja

Prošlo je dosta vremena od kada su neozbiljne spravice, po imenu „mikrodrajv" putem pošte za 28 dana, počele da stižu najhrabrijim „spektrumovcima". Neki još uvek sumnjivo vrte glavama, drugi sa grozo, a treći ih kuju u zvezde. Gde je istina, i da li je to čudo genijalnog (?) Ser Kleva stvarno četiri puta jeftinija od prave disk jedinice? Napadan i osporavan sa svih strana, mikrodrajv je ipak ušao u život mnogih ljubitelja računara. Da li je i kako položio svoj najvažniji ispit — ispit pred onima koji svakodnevno rade sa njim?

nastavimo spekulacije interfejs i sa četiri mikrodrajva daju kapacitet od 360 K za relativno skromnih 200 funti S druge strane prava disk jedinica sa kapacitetom od 200 K namenjena spectrumu košta baš isto toliko 200 funti Tu se međutim odmah postavlja jedno vrlo naugodno pitanje ako je sopstveni Sinklarov proizvod toliko nekompatibilan sa postojećim softverom sta tak treba očekivati od disk jedinice drugog proizvodača? Što se tiče kertridža iako im je cena pala sa pet funti na dve i dalje su dosta skupi Za isti novac se može kupiti diskata na koju staja 400 K podataka

U stvari na mikrodrajv sa može gledati sa nekoliko vrlo različitih pozicija

Ako koristita „spaktrum" samo da bista sa igrali onda zaboravile odmah na mikrodrajv Prabacivati igra sa kasala na mikrodrajv nija nimalo jednostavan posao i zahtjeva dobro poznavanje programiranja Uz to kasetica su i suviše skupe da bi sa na njima držala igra

Ako „spaktrum koristita za naku manju obradu podataka programiranja iz hobija ili nešto slično mikrodrajv je dobrodošao i sasvim je dovoljan Mašina sa gumenom tastaturom ne zasluzuje ništa bolja

Najzad ako sta svoju mašinu opremili profi tastaturom i ako je intanzivno koristije za programiranje ili obradu podataka onda sta stavljani pred ozbiljnu dilamu mikrodrajv ili disk jadinica? Mikrodrajv boluja od sporog pristupa podacima i Sinklarovog imena na sabi a diak jadinica od softvarsko-kompatibilnih problama Šta ja bolje za vas odlučita sami

Postoji još jadna tipično YU situacija volita kompjutare ali namata dovoljno dolara za IBM PC pa sta morali da se zadovoljite „spaktrumom Žalita disk jedinicu ali nemata 200 funti U tom slučaju smatrajte da ja mikrodrajv fantastična zamana

Ima mana ima sporosti ima naradbi čija ja sintaksa van svake pamati ali košta svega 65 funti i nauporadivo ja bolji i brži od svakog kasetofona Ako aa silom finansijskih prilika odlučita za mikrodrajv načeta sa kajati

Vladimir Kostic



Strah od kidanja Mikrodrajv predstavlja prelazni oblik između kasetofona i disk jedinice i mnogo bolji pouzdaniji od prvog i znatno lošiji od drugog sasvim je primeren računarima za koje je namenjen

ZX MICRODRIVE I INTERFACE 1

— **KAPACITET** oko 90K po kaseticl
— **PROSEČNO VREME UČITAVANJA 40K DUGAČKOG PROGRAMA** 7 do 15 sekundi
— **PROSEČNO VREME SNIMANJA 40K DUGAČKOG PROGRAMA** 15 do 20 sekundi
— **MOGUĆNOST OA SE POVEZA KAPACITET** do osam mirkodrajva ukupno 720K
— **NAREDBE ZA RAD SA MIKROORAJVOM** CAT CLOSE, ERASE FORMAT INKEY$ INPUT, LOAD, MERGE MOVE OPEN PRINT SAVE VERIFY
— **MOGUĆNOST RADA SA DATOTEKAMA** do 12 datoteka istovremano samo sakvencialne
— **RS 232** koristi nestandardan 9 polni priključak Brzina prenosa do 19200 bauda
— **LOCAL AREA NETWORK** mogućnost da sa do 64 „spektruma povežu u mrežu
— **CENA KERTRIDŽA** 2 funte
— **CENA KOMPLETA MIKRODRAJV+INTERFEJS** 1+a kasatica, 65 funti
— **ODNOS MOGUĆNOSTI/CENA** za mikrodrajv zadovoljava, za kertridže dosta slab
— **NAŠA OCENA** u nedostatku prave disk jedinice sasvim zadovoljava



My chess II 3D

MOJ ŠAH II

Verovatno još od QL-ovog trodimenzionalnog šaha čekale da se nešto slično pojavi i za komodor 64. Igranje šaha na ovom programu druga zaista ajajan vizualni utisak. Sa dva komande možete okretati tablu i zagledati je se svih strana. Za ona koji tvrde da ovakvi slakti samo smetaju kod izbijina igra šaha i traba objasniti još naka atvari. Jaks strana ova igre nije samo trodimenzionalna projekcija nago i vaoma šrrok i bogat meni. Program može da rašava problema da gra na više nivoa da vas savetuja da snimi poziciju na disk ili učita naku staru partiju ri da na printaru odštampa čitavu partiju koju ste ranije odigrali. Mogućnosti ovog šaha će sigurno obrabnti čak i one koji se užasavaju Kasparova Karpova i sličnih genijalaca da oprobaju svoju snagu bez komplakse.

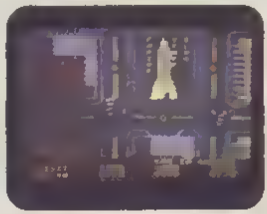


NEW YORK CITY

NJUJORK

Ova igra je napravljena da bi sve ona koji se preterano zanose željom da odu do Njujorka odvratila od takve namere aredstvima koja kao da su preuzeta iz nakog Karpanlarovog filma. Igra počinje lako što ste ispred garaže u Njujorku. Naravno u kolima ste. Pred vama je ceo grad. Čim krenete malo u grad, na vas nalati naki idiot u kolima koja su bar duplo vaca od vaših. Otpramaju vas pravo u bolnicu u kojoj morata da platite da biste izašli. Ali kola su vam sad na drugom kraju grada. Zato ulazite u banku da podignete novac koji vam je potreban ali u banci vam ne daju novac ako ne od grate jednu igru. Tu se vec naslucuje morbidni prizup autora igre. Ako budete hteli da udata u bilo koju od zgrada koje su pred vama, opet ćete morati da odigrata naku video igru. Lova koju ste uzali u banci se

polako troši i kad potpuno ostanata bez nja dobićata obaveštenja da je igra završena. Ovo ja varovatno košmarno vidanja vikenda u Njujorku za nekog jadnika iz siromašna stara Evropa: prvo vam razlupaju kola pa vas tarapu da non stop igrate video igra pa vas aultiraju kad ostanata baz love. Ako je tamo zaista tako onda je najbolje da dugo igrata ovu igru i uvažbate preživljavanje u Njujorku.



SUPER BUNNIE

SUPER ZEKA

Ovo je igra za sve one koji su odrasli sa Ouškom Dugouškom. Sadržaj je kao prenat iz nakog crtanog filma. Mali bali zeka pojede šargarapu (koja uopšte nija obična) i odjednom postaje supar zec. Igra sa sastoj iz nivoa na kojima čarobni zeka obavlja razne zadatke. Jedini problam mu je to što dajstvo čarobna šargara pa posle izvesnog vramena prastaja pa mora po novu kod veverica koja mu ja daje. Vi ćete ga naravno upitati zašto veverica na pojeda šargarapu i posiana supar ververica? Zato što veverice ne vole šargarapa ali au izgleda jako prijatajski nasatrojene prama zečavima. Posebnu slast igra predstavljaju scena kad mali zeka sraduje čudo viśta koja ga inače omalaju u dobijanju šargarapa. Možda bi Duško Dugouško to uradio drugači je ali ni ovako nija loše. Posle silnih noći igranja ćak vam ni čarobna šargarapa neća povratiti vid. Ali možda se lapiati.



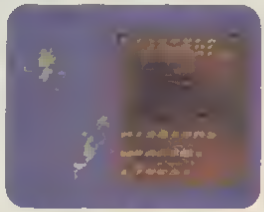
SEVEN CITIES OF GOLD

SEOAM ZLATNIH GRAOOVA

Bilo je samo pitanja vramena kad ća tvorci video igara početi da se vraćaju u prošlost i da motiva za igre vade iz atvarnih istorijskih događaja. Motiv ova igra je otkrivanje i osvajanja Amerike. Igra sadrži maltana bukvalno pranasse nu tehniku rada osvajača. Sa četiri broda i sto ljudi kraćata u osvajanje. Kada stignata do kopna i sretnata domoroce tehnika šusreta je veoma prosta — najbolje je zatražiti da vas odvedu do poglavice i tada započati borbu. Ako ubijate poglavicu ostala domoroce ćata smaći mnogo lakše i brža. Autori nisu palili od manja ulepšavanja istorije kao što vidita. Kada sa domoroci predaju pokupita im zlato napravita malo utvrđanja u kome ostavita par ljudi da se domoroci ne bi pobunili i krenete u dalja istraživanja. Tražite raka i planine i sela sa zlatom. Posle paripalija i avantura (posle) ćak i mogućnosi dezartiranja posada) sa gomilom zlata najbolja je

Pripremio: Vladimir Krstonošic

vratili se tamo odakle ste i pošli. To jest u Španiju gde ća vas kraljica prihvatili i nagradili i dali vam sradstva za novu akspedicuju. Takođe cela vidati mapu oblasti koja sta osvojili i otkrili. Prava mala imperijalistička igra.



DDNALD DUCK

PAJA PATAK

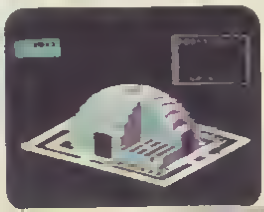
U trendu pravljanja igara po filmovima, stripovima, pa čak i knjigama bilo bi zaista čudno da je nako zaboravio na najpoznatije gundalo na svetu, Paju Patka. U ovoj igri Pajin zadatak je da nabavi igračka za svoja saatriće. Pošto nama para (igra je dakle prilagođena našim uslovima) primoran je da rad (tazgar) na četiri moguća mesta. To su železnica prodavnica farma voća i aerodrom. Količina para koju zaradi zavisi od kvaliteta obavljanog posla. Kad nakupi dovoljno para Paja pralazi prako ulice i kupuja igračke u radnjama u kojima rade Miki, Mini i Šilja. Ostatak igre je prilično jednostaven — igračka odnosa i raspoređuja po sobama a zatim dovodi jednog od sestrića da igračka isprobe. Igra dosta podseća na crtani film osim što nadostaja Pajino gundanja i napadi basa. Igrajući ovu igru sve vrame očekujata da Paja počna da se svađa sa Mikijem oko cana naka igračka. Ko zna možda u sladećoj verziji.



CRYSTAL CASTLES

KRISTALNI DVORCI

Još jedna iz dugog niza sakupljačkih igara. Vaš je zadatak da lutate kroz je šta su da su Kr stalne dvorce i da skupljate draguje koji su razbacani unaokolo. Naravno u tom vas omalaju ava moguća spodoba koje je bolesna programarska maša mogla da stvori. Daluje dosta jednostavno ali kad se doda da je sva lo u tri dimenzija kao i da postoja hodnici, liftovi i skokovi na kojima bi vam i Karl Luis pozavideo, seavim je očito da je igra zaista zanimljiva. Svi nivoi (a ima ih zaista puno) su izvanredno uradani pa vradi uložiti malo truda i par napro apavanih noći da sa preda što više nivoa. Vidi kol[illegible] može da se skupi makar i u gr



Majstorije na računaru

putovanje u provincije

Komunikacija sa periferijskim uređajima

Jedna od posebnih vrednosti računara „Komodor" je, svakako, jednostavno povezivanje i korišćenja periferijskih uređaja. Izabrano je rešenja da se operativni sistemi pojedinih uređaja ugrađuju u njih sama, pa je računaru ostalo malo posla. Zbog toga su i bejzik naredbe za komunikaciju na gotovo elementarnom nivou — samo korak dalje od onoga što je potrebno pri pisanju mašinskim jezikom. To, međutim, ne znači da upravljanja iz mašinskog jezika nema nikakvog smisla. Mašinac i ovde, kao i u mnogim drugim primenama, može značajno da unapredi performanse računara.

Za komunikaciju sa periferijskim uređajima u bejziku su nam na raspolaganju sledeće naredbe OPEN PRINT# INPUT# GET# CMD i CLOSE Primećujete da nema nikakvih specifičnih naredbi za pojedine uređaje kao na primer LPRINT ili LLIST za ispis na štampaču To je posledica činjenice da se operativni sistemi nalaze u perifernim uređajima pa računar sa svima opšti na isti način Tačnije svi uređaji priključeni preko serijskog interfejsa koriste istu rutinu računarovog operativnog sistema, pa on „ne zna koma se od njih obraća Rad ostalih (kasetofon, tastatura, RS232 i video memorija) računar programski potpuno podržava Ipak za sve je zadržan isti koncept ostvarivanja veze što ima dosta prednosti



Prva .

Da bi se razlikovali jedni od drugih, uređajima su dodaljani takozvani fizički brojevi (device naumbars) ili kako sa još nazivaju *Prve adrese*

TASTATURA	0	P
KASETOFON	1	P/S
RS 232 INTERFEJS	2	P/S
VIDEO MEMORIJA NISKE REZOLUCIJE	3	P/S
ŠTAMPAČ	4	P
DISK JEDNICA	8	P/S

U dasnoj koloni ja dat i moguć smar komunikacija (P-prijem, S-slanja)sa atanovišta računara

Pri otvaranju veza traba navsati broj uređaja koji se koriati i računar na osnovu njaga odlučuja koma da sa obrati Komunikacija sa pojedinim uređajima ja zaista jednostavna — sa tastatura se semo može dobiti informacija koji je tastar pritisnut. pa je za ovakvu komunikaciju dovoljno nsvesti broj uradaja (u ovom slučaju 0) da bi sve bilo jasno Diak jadinica možs da obavlja dalako viša poslova da formstira diskatu zapisuja i čita podatks briša pojedine programa itd Pošto se ceo oparativni sistam koji kontroliše rad diska nalazi u samoj disk jedinici javlja ae potraba da mu računar nakako stavi do znanja šta sa od njega očakuja Kako da rsepozna da li ono što mu stiže treba da zapiše kso podatak ili se radi o naredbi kada i jedno i drugo prima na isti način kao niz ASCII kodova? Zato je uvadena *Sskundarna adresa*

. . . sekundarna

Sakundarna adresa predstavlja dodatnu informaciju moglo bi se raci vid naredbe i njene su vrednosti kada ja disk u pitanju, izabrane na sledaći način 0 i 1 označavaju disku da sa radi o naradbama SAVE i LOAD koja se posebno tretiraju, 2 do 14 znači da stede podaci za datotaku dok sakundarna adresa 15 ima posabno važnu namanu Kada je primi sva no što za njom sledi disk smatra naredbom I kao takvu pokušava da ja prapozna i izvede

. . . logička adresa

Da bi sa dakla bilo šta saopštilo disku traba prvo poslati njagov broj a onda i sakundarnu adresu i to svaki put kad konatimo na primar PRINT# Moglo sa desiti da –Komodor ovde stana pa bi bajzik naredba imala sledaći oblik

PRINT#8 15 NEW naziv diskate tD

za formatiranja diskate Da aa ipak na bi svaki put pisao i broj uredaja I potrabna sakundarna adresa uvedana je nova *Logička adresa* U naredbi i OPEN kombinacji broj uredaja-sakundarna adresa dodeljuja se proizvoljan broj i ksenija samo on navodi Tako se prathodni slučaj formatiranja svodi na

OPEN 121 8 15
PRINT#121 NEW naziv diakate tD

Namarno smo izabrali , slučajan broj 121 za logičku adresu, da bismo naglasili proizvolnost njanog izbora

U komunikaciji se tastaturom ili video mamorijom nama potrebe ni za kakvim dodatnim Informacijama Pa ipak, sintaksta bajzik naredbe je za sva uređaja ista pa je moguća pisati i

OPEN 56 0,34
GET= 56 A$

ako želimo da očitamo koji je tsater pritisnut Računar će sakundarnu adresu, tamo gda ja spoljni uredaj na zahteva jednostavno ignorisati ali i dalja ostaja potrebe da se poštuja aintaksa OPEN naredbe

Moguća su i naka skraćenja u njanom pisanju Sakundarna adresa se može izostaviti, ali to nikako na znači da se ona u tom slučaju ns razmatra Jednostavno ako nija napisana računar joj daja vradnost 0 ili 255 (zašto baš ove vrednosti biće objašnjeno kasnije) Moguće ja izostaviti Čak i broj uređaja u tom slučaju se intarno postavlja vrednost 1: što znači da sa radi se kasetofonom

Postonji još jedan skraćeni oblik otvarsnja veza koji je u jednom slučaju obevezan Naima, tamo gda odmah po otvaranju treba posiati i naki ASCII niz, ns primar nseiv datotaka ili nsredbe dtsku, odgovarajućt niz se može dodstt u OPEN nsredbu Ns primar

OPEN 12,8,15
PRINT# 12 –NEW TEST

ja akvivatantno sa

OPEN 12 8 15 NEW TEST

ali je pri otvaranju datoteke na disku ili kasetofonu obavezno

OPEN 43 8 3 NAZIV S R

Parametri u OPEN naredbe mogu biti promenjive numerički izrazi i slično Tada ovako izabrano (re)tiranje periferniih uređaja dolazi do punog izražaja Uz malo pažnje pri oblikovanju ispisa (izbegavajuci TAB ili zareze) u programu se pri otvaranju veze broj uređaja može dodeliti varijabli i onda jednostavnom promenom njene vrednosti usmeravati ispis na ekran ili štampač a u jednostavnijim slučajevima i direktno u neku datoteku na disku ili traci Na primer podprogram za ispis može izgledati ovako

OPEN 10 DV SA A$
FOR F=0 TO 20 PRINT# 10 R (F) NEXTF
CLOSE 10
RETURN

Ako ga pozovete sa ulaznim parametrima DV=3 (SA i A$ mogu imati bilo koje vrednosti) rezultat će biti ispisan na ekranu Ako je DV=4 SA=0 (A$ nije vazan) rezultat ce biti smesten u sekvencijalnu datoteku na disku odakle kasnije može biti korišćen u nekom drugom programu na primer tekst procesoru i slično

Koncept se logičkom prvom i sekundarnom adresom se očigledno dopao „Komodoru pa je dosledno poštovan i na nekim mestima gde komplikuje stvari Na štampaču 1526 (MPS 802) se gotovo sve funkcije ostvaruju korišćenjem odgovarajućih sekundarnih adresa (čak 11) iako je isto moglo biti postignuto primenom nekoliko kontrolnih karaktera Tako se izbor seta velika/mala slova vrši otvaranjem veze sa sekundarnom adresom 7 ali se isto može postici i bez njenog navođenja ako se pre stringa pošalje karakter 17 (kursor dole)!

Sistemske promenljive

Za rad se periferijama rezervisan je veliki broj sistamskih promenjivih u nultoj i prvoj strani memorije Mnoge od njih su interni — brojači pokazivači, zastavice ili beferi i nisu od značaja — za korisnika ostaje svega nekoliko važnih koje ćemo nabrojeti

152 — broj otvorenih veza
153 — ulazna jedinica
154 izlazna jedinica

183 dužina naziva datoteke
184 — logička adresa
185 — sekunderna adresa
186 - prva adrasa (broj uređaja)
187 i
188 — pročetna adresa naziva datoteke

601 do 610 — tablica logičkih edresa
611 do 620 — tablica prvih adresa
621 do 630 — tablica sekundarnih adresa

Kada zadamo OPEN računar uzima prvi parametar iza naredbe i njegovu vrednost dodeljuje logičkoj adresi Zatim se broj uređaja postavlja na 1 (kasetofon) sekundarna adresa i dužina naziva datoteke na 0 Sada se tek ispituje (odatle mogucnost skracivanja) da li je zadat sledeći parametar (broj uređaja) i ako jeste ranije dodeljena vrednost 1 se menja u zadatu Isto važi i za sledeci parametar kao i za string na kraju s tim što ukoliko nema sekundarne adrese a broj uređaja je veci od 3 ranije postavljena vrednost 0 se menja u 255 Sve ove vrednosti se smeštaju u sistemske promenjive 183—188

Sada se poziva rutina koja će pretražiti tablicu logičkih adresa da bi se utvrdilo da li je vec ranije otvorene veza se istim brojem i ako jeste izdala poruka FILE OPEN ERROR Ako nije logička prva i sekunderne adrese se ubacuju na krajeve odgovarajucih tablica i broj otvorenih veza (152) povecava za jedan Time je veza otvorena i dalji tok dogadaja zavisi od broja uređaja

Najčešce se neće desiti ništa vise Tako se nakon OPEN 4 4 i ako stampac nije priključen ne dobija poruka DEVICE NOT PRESENT U slučaju disk jedinice odmah se ako postoji šalje string iz OPEN naredbe koji predstavlja naziv datoteke ili neku od naredbi

Dve sistemske promenjive (153,154) su posebno značajne Na adresi 154 se nalazi broj uređaja na koji je u tom trenutku usmeren prenos podataka iz računara Ako pogledate njen sadržaj, videćete da se tamo nalazi broj 3 što znači da je ispis u tom trenutku usmeren na ekran Šta se dešave kada nakon otvaranja veze upotrebite naredbu PRINT#? Računar pretražuje tablicu logičkih adresa i kad nade onu koju ste naveli u naredbi iz ostale dve tablice uzima odgovarajuću prvu i sekundarnu adresu r sve to prebacuje u sistemske promenjive 184 185 i 186 Zatim uređaju naređuje da počne da sluša i ako je potrebno šalje sekundarnu adresu Vrednost iz adrese 186 prebacuje u 154 i vrši skok na PRINT rutinu, istu onu koju koristi i „običan PRINT Kada je i poslednji bajt poslat, spoljnom uređaju se naređuje da prestane sa slušanjem i vrednost sistemske promenjive 154 se vraca na 3 Sem u jednom sučaju ako je upotrebljena naredba CMD- Njenim zadavanjem nakon OPEN vrednost sistemske promenjive 154 se postavlja kao u gore opisanoj proceduri, ali na toj vrednosti i ostaje Sada i „običan PRINT ili LIST umesto na ekran šalje ispis na spoljni uređaj Ovakvo stanje ce trajati sve dok se preko veze ne pošalje jedan Prazan PRINT#

Dakle rutina koja vrši slanje karaktera ispituje jedino vrednost u lokaciji 154 da bi ustanovila o kom se uređaju radi Slične procedure se odvija i pri toku informacije od periferije ka računaru — tada je važna sistemska promenjiva 153 s tim što u bejziku ne postoji ekvivalent za CMD u slučaju INPUT naredbe

Komunikacija iz mašinca

Retko se pruže prilika da pisanje u mašinskom jeziku bude jednostavno kao u bejziku U radu s spoljnim uređajima upravo je to slučaj Sve naredbe bejzika se jednostavno zamenjuju pozivom jednog ili najviše tri potprograme Usvojene je i sledeca konvencija koja je dosledno poštovana

Svi potprogrami se pozivaju indirektno preko tzkozvane KERNAL JUMP tablice čime se obezbeduje nezavisnost mašinskog programa od verzije operativnog sistema računara (broj verzije možete saznati upotrebom PRINTPEEK (65408)

• Pri povratku iz potprograma CARRY zastavica ako je 0, označeva da je sve proteklo u redu U suprotnom u akumulatoru ce se naći kod greške od 0 do 9 sa sledecim značenjem

0 — rutina prekinuta STOP tasterom
1 — previše otvorenih veza (više od 10)
2 — veze vec otvorena
3 — veza nije otvorena
4 — datoteka nije naden
5 — uređaj nije priključen
6 — uređaj nije predviden za slanje podataka
7 — uređaj nije predviden za primanje podalaka
8 — nedostaje naziv datoteke
9 — nepostojeći broj uređaja (device number)

Interesantno je da je Komodor predvideo mogucnost i da se broj greške u trenutku nastanka odštampa na ekranu Da bi ovo omogucili treba šesti bit u sistemskoj promenjivoj na adresi 157 postaviti na 1 Bejzik ga pri svakom prelasku u direktan mod vraca na nulu ali probajte u jednoj liniji

POKE157 192 PRINT# 43

Iz navedenog spiska se vidi da se greške ugtavnom odnose na samo otvaranje veze i pripremne rutine Informaciju o stanju u toku samog slanja ili primanja podataka dobijamo iz sistamske promenjive STATUS na adresi 144 Postoji i potprogram koji očitava njenu vrednost i vraca je u akumulatoru Iako je on krajnje jednostavan, zapravo izvodi jedino instrukciju LDA 144 ipak se preporučuje da ga koristite umesto direktnog očitavanja memonje zbog kompatibitnosti različitih verzije Komodor računara

Svaki bit STATUSA ima posebno značenje od kojih su praktično samo '3 od važnosti Ako je neki od njih 1 znači da je

bit 5 — greška u kontrolnoj sumi pri čitanju sa trake
bit 6 — primljen karakter poslednji u datoteci
bit 7 — uređaj nije priključen ili je našao kraj trake

Bez obzira na ovako razgranat sistem obaveštavanja o greškama on nije dovoljan da pokrije i disk jedinicu Iz STATUSA se samo dobija informacija da nešto nije u redu, nekad ni to, i da bi se ustanovilo koja je stvarno greska nastala potrebno je posebno očitavanje komandnog kanala

U prilogu 1 dat je program koji uvodi novu naredbu LPRINT koja je standardna u većini bejzika Ima istu namenu kao i PRINT ali ispis ide na štampač Iako na komodoru za njom ne postoji potreba koristi se PRINT# namerno smo izabrali ovaj primer da bi ilustrovali šta je to „Komodor trebalo da dopiše u bejzik interpreteru da bi oslobodio korisnike razmišljanja o otvaranju veze i sekundarnoj adresi Ujedno ćemo na tom primeru objasniti i pojedine potprograme operativnog sistema zaduženo za komunikaciju Dodavanje nove naredba izvedeno je na način objašnjen u „Računarima 6 pa ga ovde nećemo obrazlagati

De bi se otvorila veza potrebno je prvo postaviti odgovarajuće parametre Ovo će obaviti dve potprograma, SETLFS (sat logical first and secondary addres) i SETNAM (set name) koji se pozivaju na sledeći način

A = logička adresa
X = broj uređaja (prva adresa)
Y = sekundarna adresa
JSR SETLFS
A - dužina naziva
XY = početna adresa naziva (X niži bajt)
JSR SETNAM
U našem primeru nama potrebe za nazivom datoteke pa je rutina koja ga postavlja pozvana sa nulom u A. Svejedno koju čete od navedene dve rutina prvo pozvati. Važno je samo da su obe izvedene pre aktiviranja OPEN-a to je OPEN. Nakon njega veza je otvorena na identičan način kao da je u bejziku izvedeno OPEN 10,4,0. Primećujete da nismo testirali CARRY zastavicu. Pretpostavili smo da se nece desiti da je negde u programu pre izvođenja naredbe LPRINT već otvorena veza sa logičkom adresom 100. što je praktično jedina greška koja bi u ovom trenutku mogla nastati (ako ste to ipak učinili dešavaće se neobične stvari).

Veza je neaktivna sve dok se ne zada smer komunikacije. U bejziku se pri izvođenju PRINT# naredbe veza automatski proglašava izlaznom, ali u mašinskom programu se to mora posebno izvesti. Za to služi rutina sa ulaznim parametrom u X registru.

X = logički broj veze
JSR CHKOUT

Dejstvo ovog potprograma je slično kao naredbe CMD u bejziku (CMD se u suštini lokalno izvodi pre svakog PRINT# ali se odmah po ispisu njeno dejstvo prekida). Sada je ispis stalno usmeren na štampač i to će potrajati dok se ne izvede rutina

CLRCHN (clear I/O channels)

Njen zadatak je da komanduje svim uređajima da prestanu sa slušanjem i da vrednost sistemske promenjive 154 vrati na 3 (za ispis na ekran). Dakle treba je izvesti odmah po završetku slanja ili prijema. Ako to ne učinimo već otvorimo novu vezu za slanje, podaci će ići na obe. Ovo nije greška već predviđena mogućnost koja otvara interesantna primena ali važi samo za uređaje priključene preko serijskog interfejsa.

CLRCHN na zatvara vezu. Nakon ove rutina stanja je isto kao i odmah nakon OPEN. Da bi novi podaci bili poslati preko iste veze dovoljno je prethodno izvesti samo CHKOUT.

Još jednom o greškama. U programu amo, nakon CHKOUT ipak ispitali CARRY zastavicu zbog toga što je to prva rutina po otvaranju koja stvarno i dovodi računar u vezu sa periferijom (ako nema naziva datoteke OPEN završava u samom računaru). Tada je tek moguće ustanoviti da li je uređaj priključen pa je u tu svrhu testiranja i izvršeno. Ako nije, u akumuletoru će se naći broj 5 što je igrom slučaja i broj bejzik poruke DEVICE NOT PRESENT ERROR. Igra slučaja važi i za ostalih 8 mogućih grešaka! Iseno ovo nije slučajnost ali ipak treba imati u vidu da se radi o odvojenim porukama. Ovako izabrani brojevi grešaka tačnije raapred bejzik grešaka, omogućuje da se lakoriati i bajzik deo kako je to u ovom primeru i urađeno (X = broj bejzik greške pa JSR 42039).

Zatvaranje veze se izvodi jednostavno se

```
PRILOG 2

a
40158                           .OPT P4
40158                           *=   40158
45192                   SETNAM  =    65465
45192                   SETLFS  =    65466
45192                   OPEN    =    65472
40152                   CHKIN   =    65478
45192                   CHKOUT  =    65481
45192                   CHRIN   =    65487
40152                   OETIN   =    65508
45192                   CHROUT  =    65488
45158                   CLRCHN  =    65484
45192                   CLOSE   =    65475
45152                   READOT  =    65463
40158                   INPUT   =    42336
45158                   PRINT   =    43886
                ;////////////////////////
                ;/  IZMENA BWB VEKTORA  /
                ;////////////////////////
49152  160 811                  LOA  R<NEWBWS
49154  162 192                  LDX  R>NEWBWS
45156  141 002 003              STA  770
45155  142 003 003              STX  771
49162  096                      RTS
                ;////////////////////////
                ;/     NOVI BWB       /
                ;////////////////////////
49163  032 806 165    NEWBWS    JSR  INPUT
49166  134 122                  STX  122
49168  132 123                  STY  123
49170  032 115 000              JSR  115
49173  170                      TAX
49174  240 043                  BEQ  NEWBWS
49176  201 033                  CMP  R"!"
49178  240 006                  BEQ  DISKCOM
40180  032 121 000              JSR  121
49183  076 144 164              JMP  42120
                ;////////////////////////
                ;/    OISK NAREOBE      /
                ;////////////////////////
49186  169 255        DISKCOM   LOA  #255
49188  133 058                  STA  58
49190  032 115 000              JSR  115
49193  201 036                  CMP  R"@"
49195  200 003                  BNE  CP1
49197  076 077 192              JMP  OIRCT
49200  201 033          CP1     CMP  R"S"
49202  208 003                  BNE  OSTL
49204  076 110 192              JMP  SCRATCH
                ;////////////////////////
                ;/ OSTALE DISK NAREDBE  /
                ;////////////////////////
49207  032 253 192    OSTL      JSR  OPEERR
49210  162 099                  LDX  R99
49212  056                      SEC
49213  032 040 193              JSR  OIREC
49216  169 001                  LOA  R1
49218  160 002                  LOY  #2
49220  032 030 171              JSR  PRINT
49223  032 111 193              JSR  READER
49226  076 134 227              JMP  50246
                ;////////////////////////
                ;/ OIREKTORIJ OISKETE   /
                ;////////////////////////
49220  032 253 192    OIRCT     JSR  OPEERR
49232  032 014 193              JSR  OPOIR
49235  162 098                  LOX  R98
49237  024                      CLC
49238  032 040 193              JSR  OIREC
49241  032 058 193              JSR  GETBYT
49244  032 058 193              JSR  GETBYT
49247  032 020 103      DRO     JSR  OETLIN
49250  169 013                  LOA  #13
49252  032 210 255              JSR  CHROUT
49255  173 141 002      WAIT    LOA  653
49258  200 251                  BNE  WAIT
49260  240 241                  BEQ  DRO
                ;////////////////////////
                ;/  BRISANJE FAJLA     /
                ;////////////////////////
49202  032 115 000    SCRATCH   JSR  115
49265  170                      TAX
49266  240 155                  BEQ  OSTL
49268  201 042                  CMP  R"*"
45270  200 246                  BNE  SCRATCH
49272  032 253 192              JSR  OPEERR
49275  169 147          AGAIN   LOA  #"J"
46277  032 210 295              JSR  CHROUT
46280  032 014 193              JSR  OPOIR
49283  162 098                  LOX  R98
49285  024                      CLC
49286  032 040 103              JSR  DIREC
```

```
PRILOG 1

E                                       OBJ P4
49152                                   4K  49152
49152           SETNAM  =       $FFBD       49152
49152           SETLFS  =       $FFBA       00680
49152           OPEN    =       $FFC0       00682
49152           CHKOUT  =       $FFC9       49478
49152           CLRCHN  =       $FFCC       00681
49152           CLOSE   =       $FFC3       00684
49152                                       49470
                ;/////////////////////////////
                ;/  IZMENJK BCD VERTORA     /
                ;/////////////////////////////
49152   18E 011                 LDA  #11NEWBCD
49154   182 180                 LDX  #$NEWBCD
49156   141 000 003             STA  778
49159   14E 000 003             STX  777
49162   096                     RTS
                ;/////////////////////////////
                ;/  NOVA BCD RUTINK         /
                ;/////////////////////////////
49163   E32 118 006   NEWBCD    JSR  115
49166   0C1 07F                 CMP  #"L"
49168   E40 006                 BEQ  NEXTCHK
49170   032 1E1 006   BLANCK    JSR  1E1
49173   07F 2E1 187             JMP  $2E1B3
49176   182 021       NEXTCHK   LDY  $1
49178   177 1EE                 LDA  11EE1 Y
49180   0D1 182                 CMP  $1E2
49182   109 E4E                 BNE  BLANCK
49184   032 110 006             JSR  115
49187   082 11E 006             JMP  110
49190   006                     CHP
49191   B7B                     PHA
                ;/////////////////////////////
                ;/  NAREDBA LPRINT          /
                ;/////////////////////////////
                ;
                ;LBK. PRV  I BER   ADRESA
                ;
49194   1B0 186                 LDA  #186
49196   182 004                 LDX  #4
49198   18E 000 006             LDY  #0
49182   09E 180 2B0             JMP  $B7LF8
                ;
                ;INE
                ;
49401   1B0 09E                 LDA  #9
49203   032 180 250             JMP  $2TNAM
                ;
                ;OTVARANJE VEEE
                ;
49206   032 1EE 250             JMP  $EEN
49209   182 182                 LDX  #182
49211   032 E01 2C0             JSR  CHKOUT
49214   144 006                 BCC  $YART
                ;
                ;AKO JE NASTALA GREBKA
                ;
49218   178                     TAX
49217   07B 0B0 194             JMP  $B0B2
                ;
                ;VRACANJE PRVOG ZNAKK ER STAMPANJE
                ;PR 00GIC PRINT RUTINA
                ;
49220   184             $YART   PLA
49221   006                     ZLZ
49222   032 182 178             JMP  $J000
                ;
                ;KRAJ ETAMPANJA  ZSTVARANJE VEEE
                ;
49225   032 E64 255     EXIT    JSR  CLRCHN
49228   180 180                 LDA  #180
49230   032 180 2B0             JMP  CLOSE
49233   07B 174 107             JMP  $E3E8
                ;
49202   0B8 005                 BVC  $YART
49204   178 178 180             PLP
                ;/////////////////////////////
                ;/ SETALE BIBR NAREBBE      /
                ;/////////////////////////////
49207   03E 09E 182     GETL    JSR  $9EEPR
49210   18E 000                 LDX  #00
49212   032                     BEC
49213   0E 048 183              JBO  $13EC
49218   185 001                 LDA  $1
49218   18E                     LDY  EE
49220   032 038 171             JBO  PRINT
49223   03E 111 188             JSR  $EINE
49226   07B 134 187             JBE  $5B2A0
                ;/////////////////////////////
                ;/ DIREKTORIJ BISKETE       /
                ;/////////////////////////////
49229   E32 253 192     $IRCT   JSR  $9EERR
49232   032 014 193             JSR  $PDIR
49235   18E                     LDX  #88
49237   B24                     CLC
49238   032 048 193             JBO  $1REC
49241   032 0B6 183             JBO  $ETBYT
49244   03C E38 193             JBO  $ETBYT
49247   032 08E 193     000     JSR  $ETLIN
49250   18E 000                 LDX  #12
49252   0JE 21B 255             JBO  CHRNUT
49255   173 141 006     WAIT    LDA  $53
49258   248 2A1                 BNE  WAIT
49260   248 2A1                 BEQ  000
                ;/////////////////////////////
                ;/  BRISANJE FAJLA          /
                ;/////////////////////////////
49262   B3C 11E 006     SCRATCH JSR  115
49265   178                     TAX
49266   248 155                 BEQ  $STL
49268   281 042                 CMP  #"*
49270   288 248                 BNE  $CRATCH
49272   B4C 255 100             JSR  $PEERR
49275   158 147         AGAIN   LDA  #"J"
49277   B3C 81B 255             JSR  CHRNUT
49280   B3C 814 193             JSR  OPDIR
49283   162 098                 LDX  #98
49285   024                     CLC
49286   B3C 048 153             JSR  $1REC
49289   B3C 0B6 193             JSR  $ETBYT
49290   B3C 0B6 193             ISR  $ETBYT
```



```
49293   B3C 08E 193             JSR  $ETLIN
49296   162 813                 LDX  #13
49298   B3C 218 255             JSR  CHRNUT
49301   B32 08B 183     RSK     JSR  $ETLIN
49304   0E 225                  INC  211
49306   0JB 211                 LDX  211
49308   1B6 211                 CPX  432
49311   224 B3C                 BCC  $05
49313   144 248                 LDA  # 38
49315   1A8 B38                 JSR  CHRNUT
49318   B3C 218 255             JSR  CHRNUT
49321   B3C 2B4 255             JSR  CLRCHN
49324   B4C 226 255     GETKEY  JBO  GETIN
49327   248 251                 BNE  $EIKEY
49329   17E                     TAX
49330   201 B4B                 CMP  #"@
49332   248 B2C                 BNE  MBB
49334   032 B7B                 LDA  #"M"
49336   03C 218 255     MBB     JBO  CHROUT
49339   169 B1E                 LDA  #13
49341   03C 218 E55             JSR  CHROUT
49344   224 B4B                 CPX  #"@
49346   288 B5B                 BNE  NEXT
49348   169 145                 LDA  #145
49350   03C 218 255             JBO  CHROUT
49353   169 B0B                 LDA  #$0B
49355   032 183 E58             JSR  CLOSE
49358   1B6 B3                  LDX  #3
49360   133 153                 STA  153
49362   1B0 B02                 LDA  #$02
49364   141 B03 006             STA  $15
49367   1B0 B0B                 LDA  #"1"
49369   141 B02 006             STA  $10
49372   194 B0B                 LDY  #$0
49374   0C4                     DEC
49375   194 B0B                 LDA  #$0
49377   194 B0B                 LDY  #$0
49379   006                     RTS
49380   B0B 0AB 183             JBO  $1REC
49383   1E0 B02                 LDX  #3
49385   1B0 006                 LDY  #0
49387   B0B 170 171             JSR  PRINT
49390   032 006 E55             JSR  CLRCHN
49393   070 183 1B0             JMP  AGAIN
49396   162 006         NEXT    LDX  #$0B
49398   B54                     CLC
49399   006 006 1B8             JBO  $1REC
49402   070 151 182             JBE  ABK
                ;
                ;OTVARANJE ERRER KANALA
                ;
49405   1B0 B0B         PREERR  LDA  #$0
49407   180 008                 LDX  #8
49409   1B0 B15                 LDY  #15
49411   032 1B0 295             JBO  $BTLF8
49414   1B0 BAB                 LDA  #$0
49416   153 153                 STA  153
49418   032 1B0 B0B             JBO  OPEN
49421   006                     RTS
                ;
                ; OTVARANJE DIR. ZNJLA
                ;
49422   1B0 006         PBDIR   LDA  #$0
49424   1B0 B0B                 LDX  #2
49426   160 006                 LDY  #0
49428   032 1B0 285             JBO  $BTLF8
49431   1B0 182                 LDA  #"$"
49433   153 006                 STA  2
49435   1B0 B01                 LDA  #1
49437   182 B0B                 LDX  #2
49439   168 006                 LDY  #0
49441   032 1B0 2B0             JBO  $BTNAM
49444   B3B 1B2 B05             JBO  OPEN
49447   006                     RTS
                ;
                ;IZBOR BHERR KBMUNIKACIJE
                ;
49448   144 006         $IREC   BCC  IN
49450   B32 B01 B05             JBO  CHKOUT
49453   178 B07                 BCS  $RR
49455   07B                     RTS
49456   032 182 B05     IN      JBO  CHRIN
49459   178 B01                 BCS  ERR
49461   006                     RTS
49462   178             ERR     TAX
49463   07B 685 184             JBE  42B3B
                ;
                ; UZIMANJE 1 BYTE-R IE RNJLA
                ;
49466   006 E07 E05     GETBYT  JSR  CHPIN
49469   178                     TAX
49470   032 183 29E             JBO  $E006T
49473   1B0 B0B                 BNE  ENB
49475   159                     TXA
49476   006                     RTS
49477   006 111 18E     ENR     JSR  $ERR2B
49480   1B0 B9B                 LDA  E08
49482   B0B 185 E0B             JBO  CLOSE
49485   B70 194 E07             JBE  $B6AB
                ;
                ; CITANJE I STAMPANJE 1 NNEIVR
                ;
49488   B32 B5B 1B8     GETLIN  JBO  $ETBYT
49491   B32 B5B 193             JSR  $ETBYT
49494   B0B 1B4 19E             JSR  $ETBYT
49497   072                     PHA
49498   B32 B5B 1BE             JSR  $ETBYT
49501   1AB                     TAY
49502   104                     PLA
49503   178                     TAX
49504   098                     TYA
49505   B32 B05 1B0     PRT     JBO  40B0B
49508   B3C B02 1B0     PRT     JBO  $ETBYT
49511   E43 B0B                 BCC  $RTB
49513   B88 21B 255             JSR  CHROUT
49516   144 B4B                 BCC  PRT
49518   006             RINIBK  RTB
                ;
                ; OCITAVANJE I ZRTV. ERRER KANALA
                ;
49519   B32 B04 B05     READER  JBO  CLRCHN
49522   1B0 B04                 LDA  #$0
49524   B64                     CLC
49525   2B0 B4B 183             JBO  $1BEC
49528   B32 B0B 1B5             JBO  INPUT
49531   ABK B04 B55             JBE  CLRCHN
49534   1B0 B0B                 LDA  E$0
49536   2B0 1B0 B05             JBO  CLOSE
49539   1B9 B0B                 LDA  #$0
49541   1B0 B02                 LDY  #$0
49543   032 B5B 171             JBE  PRINT
49546   006                     RTS
```

A = logički broj veze
JSR CLOSE

Nešto bolji DDS

Primer 2 je nešto složeniji i ilustruje rad sa disk jedinicom kao i rad sa tastaturom i video memorijom niska razolucija kao paritarnim uređajima. Program bi se uslovno mogao nazvati DOS jer omogucava korišćenje stvarnih DOS naredbi (koje sa prepoznaju i izvode u samoj disk jedinici) na isti način na koji sa korista i bejzik naredbe izmana BWS vaktora omogucava rad programa laključivo u direktnom modu

Prvi dao (NEWBWS) je prepisan iz bajzik ROM-a Linija koju otkucate do pritiska na tastar RETURN naći ća se u INPUT bafaru od adrese 512 Ako sa na prvom mestu, na računajući razmake o čemu sa brine JSR 115 nede znak ugtaste zagrada program nastavlja sa radom U suprotnom se izvršavanje prepušta bejziku (tokenizovanje naredbi i izvršavanja ili smeštanja linija u program)

Dodali smo i dve opcija koje inaća nisu na respolaganju Ako otkucate 1$ na akremu će biti izlistan direktorij baz upisivanje u mamoriju računara Ako želita da obrišate maku datoteku sa diskate pa zadate 1S " disk bi shvatio da traba obrisati sva U tom slučaju program naća odmah proaleditи naredbu već ća listati direktorij jedan po jedan ulaz i tražiti da pritisnata D za brisanja ili N ako brisanja nije potrabno Tako sa viša datoteka može brže i jednostavnija izbrisati nago stalnim ponavljenjam naredbe

U svakom drugom slučaju, bez obzira šta napišata iza zagrada string se prosleduje disku na obradu Ovaj posao obavlja deo od labela OSTL Otvara sa komandni kanal (sakundarna adresa 15) i proglašava izlaznim Naredba aa šalja rutinom iz bejzik ROM-a (43806) čiji je ulazni paramatar počatna adresa atringa u A i X ragistru (A niži bajt) Kao terminator služi 0 ili 34 (znak navoda) Ovda je upotrabljana 0 koju rutina INPUT (42336) automatski postavlja za ulazni (uz prethodno CLRCHN) i očitava poruka sa diska (potprogram READER)

Moguća je otkucati samo zagradu Tada se disku ne šalje ništa već samo očitava komandni kanal što može biti korisno kad u toku rada crvana LED dioda iz sasvim narazumljivih razloga počna da trapća

Diraktorij diska je posabna datoteka sa praciznim formatom ali ja u DOS ugradana i mogućnost njanog tretiranja kao bilo kog programa Traba ga otvoriti sa sakundarnom adraaom 0 (u počatku je rečano da je ona razarvisana za LOAD naredbu) i disk ća ga posiati u obliku koji odgovara programskoj datotaci Prva dva bajta predstavljaju adresu od koje se program smešta u mamoriju ako sa upisuja sa LOAD ima 81 Zatim za svaki ulaz sleda dva bajta takozvana vaza sa sladećom linijom i još dva koja predstavljaju linijski broj u ovom slučaju broj blokova koje datoteka zauzima Zatim sledi naziv i bajt nule kao tarminator linija

Otvaranja diraktorije je napisano kao potprogram OPDIR Kao naziv potraben ja samo znak $ pa smo ga smastili na adresu 2 (potpuno proizvoljno) Uzimanje jednog bajta sa vaze povarili smo potprogramu GETBYT On koristi rutinu operativnog sistema CHRIN (input charactar) i odmah provarava stanja STATUS varijabla U slu-
```
```

čaju da je vrednost različita od nule znači da je uzet poslednji bajt iz detoteke, ili je nastala neka greške u čitanju sa diska Tada se dalje čitanje prekida očitavc komandni kenel de bi se dobila poruke o stvarnoj greški i program prekida se radom JMP 58246 je skok ne takozvanu ‚farmu za inicijalizaciju steka ispis READY itd

Čitenje bilo koje sekvencijalne ili programske detoteke se izvodi ne identičen nečin Tada je nepotrebne rutine GETLIN u kojoj se dva bajta (broj zauzetih blokova) štampaju kao string

Opcija za brisanje detoteka otvara direktorij i uzima njegov sadržaj ne prethodno opisan nečin Brisanje ekrana se postiže slanjem keraktera 147 Rutina

A = bajt
JSR CHROUT

šalje jedan bajt ne uređaj koji je u tom trenutku proglašen otvoranim ze ispis Kako to u ovom slučaju nije uređeno prethodno ni za jednu vezu tapis je usmeren ne video mamoriju

Kada sa ispiše naziv jedne detoteke Ireba očitati tastaturu de bi se ustanovilo koji je taster pritisnut Ovo se postiže pozivom rutine GETIN koje u ekumulatoru vraca kod pritisnutog tastere Ako nijedan nije pritisnut vrednost će biti 0 Primecujete da smo prethodno resetovali sve ulazne i izlazne veza sa CLRCHN To nije uređeno slučajno jer je rutina GETIN dizajnirana za čitanje ne samo tasteture već i bilo koje veze koje je otvorena za upis Istu nemenu ima i ranije pomenuta CHRIN U čamu je onde razlika?

GETIN je preveshodno nemanjana tastaturi RS232 interfajsu i uzime samo jeden bajt Keda tasteluru očitevete CHRIN rutinom na ekrenu se pojavljuje kursor i izvodi isto što i zedavanjem bajzik INPUT naredbe Tek ked detektuje taster RETURN, računer napuste ovu rutinu i vreča kod znake koji je prvi otkucan Sledeći poziv uzima redom dalje otkucane znake i ked su svi obrađani vreča bajt 13 Novi poziv ponavlje proceduru Upotreba GETIN umseto CHRIN za ostele uređaje (osim RS232) ne previ razliku

Ako ste pritisnuli tester D potrabno je de se disku pošalje neredba

S naziv datoteke

Kako dobiti naziv detoteke kade je jedino ispisan na ekrenu? Uzimenje direktno iz video memorije stvera doate probleme jer Ireba ispiteti ne kojoj linij se naziv nalazi i ujedno, ekranske kodove prevesti u ASCII Rešenje je jednostavno jer Komodor i vidao memoriju tretira kao periferni uređaj INPUT se ekrene će smsatiti u input beter sadržaj jedne linije od trenutnog potoženje kursora do kreje reda Ili znekova „ I Pri loma se početna prazne meste Ignorišu (INPUT prazna linije vreča prazen string)

Da bismo otvorili vezu se video memorijom i proglasili je otvorenom ze upis upotrebili smo nedozvoljenu proceduru nismo je ni otvarali, niemo pozivali CHKIN već smo sistemskoj promenjivoj 153 (broj ulazna jedinice) dodelili vrednost 3 Ovo je jedini sučaj u kome je ovakva procedure moguča — CHKIN u tom slučeju uredi potpuno isto, eli smo ovej nečin deli redi ilustrecije e ne keo preporuku Uštede od 14 bejtove koje se njom postiže nije adakvetne gubitku razumljivosti vešeg programe

Postavljenje kursora ne početak linije smo obazbedili iepisom karektere 13 (novi red) i 145 (kursor gore, U input beter ee dodaje slovo S i prvi znak navode zamenjuje sa

Kede je naredbe izvršene direktorij more ponovo de se čite od početka jer smo mu brisenjem jedne detoteke izmenili sedržaj Ovo sa može izbeći prethodnim upisivenjem celog direktorije u memoriju, pa tek onde njegovom obredom što bi zahtevelo odvajenja memorijakog prostore (u nejgorem slučeju oko 3 K) ili upisom direktorije u memoriju blok po blok kao RANDOM detoteke uz vodanja računa o formatu u kome su ulazi zapisani

Šta je se naredbeme SAVE i LOAD? Njihovo izvršenje je povereno posebnim rutinneme operativnog sistema se islim nazivima De bi segment memorije snimili ne diskatu Ireba već pomenutim rutiname SETNAM i SETLFS postaviti naziv i logičku prvu i sekundernu adresu Zelim se poziva SAVE (adresa 65496) ne sledeći nečin

A = adrese u nultoj streni i ne kojoj se zajedno se sledećom nelazi adrese prvog bajta koji treba snimiti
XY = edresa poslednjeg bajte (X niži bajt)
JSR SAVE

Potreba za SETLFS u ovom slučeju može da zbuni SAVE rutine uopšte ne uzime u obzir logički broj a u slučeju diska ni sekundernu adresu, već je uvek postavlje na 1 Zeto u bajziku neme baš nikekve razlike u oblicima SAVE „ime 6 ili SAVE „ime 8 1 Sa kesetotonom je situecije nešto drugačije Ako se snimanje izvrši sa sekundarnom adreeom 1 program će kesnije pri LOAD uvek biti smešten na isto meeto se koge je i snimljen bez obzira na sekundernu adresu u LOAD neredbi Ako upotrebimo vrednosi 2 posle programa ce biti zepisano još jedno zaglevlje kao oznaka kraj Ireke Sekundarne edresa 3 kombinuje ove dve osobine

Rutine LOAD (edrese 65493) je kombinovana se VERIFY i zehteve postavljanje parametere sa SETNAM i SETLFS Poziva se sa

A = 0 za LOAD itl
A = 1 za VERIFY
XY = početne edresa
JSR LOAD

U ovom slučeju sekunderne adrese ime svoj amisao Ako je 1 nije potrebno de u XY registrima navodite početnu adresu Biće iskorišćene ona se koja je program snimljen Ako je 0 onde se obevezno zadaje početne edresa što omogućuja ralociren upis

Načuli uvo i slušaj

U prethodnim primerime doata smo koristili dve rutine iz bajzik ROM-e INPUT i PRINT Obe u sebi sedrže petlja unutar kojih au CHRIN i CHROUT. U većini programe ćete sresti slučej da eu autori sami dopisivali slične potprograme Ovo je svakako preporučljivije iz prostog razloge što ovakvi programi ne treba de se oslenjeju na pomoč bilo kog jezičkog interpretere de bi radili sa svakim od njih Ujedno, čest je slučaj da se bajzik ROM izbecuje iz edreenog prostore mikroprocesore i za program koristi RAM ‚ispod njega.

Pažljivi čitelec je sigurno primatio de smo ne nekoliko meste u ovom tekstu upotrebiti izraza komenduje spoljnom uredeju da sluša i slične bez objašnjenja Ovo je specifičnoat uradeje priključenih preko serijskog interfejse eli se primećuje samo lokalno kede se kanel proglešava otvorenim ze ispis korisniku je avejedno o kom se uredeju redi uvek se koristi ista rutine CHROUT Ze štampač III dlek, računer preduzime daleko više radnji nego što je to slučej ze tasteturu ili video memoriju Tako dolazimo do jednog još elemantarnijeg nivoe komunikacija sa ovim uredejime koji je u komercijelnim programime dosta časi

Sve rutine ze komunikeciju sa ovim uredajime koriste 8 oanovnih

LISTEN (65457) — nareduje spoljnom uredaju da preda na prijem (da „sluša)
SECOND (65427) — šalja sekundarnu adresu nakon LISTEN
CIOUT (65448) — šelja jedan bajt
UNLSEN (65454) — komenduja evim uredejima de prestenu se prijemom
TALK (65460) — komanduje spoljnjem uredeju da počne sa slenjem podateka
TKSA (65430) — šalje sekundernu adresu nakon TALK
ACPTR (65445) — prima jeden bajt
UNTLK (65451) — komenduje svim uredejima da prestanu sa slenjem

Da bi se na primer odštempeo neki znek na šlepaču bio bi potreban sledeci program

```
LDA # 4
JSR LISTEN
LDA # 96
JSR SECOND
LDA # bajt
JSR CIOUT
JSR UNLSN
```

Prvo što se primećuje je neobične sakunderne edresa. Njena vrednost je, u stvari 0 eli su joj bit 6 i 5 setoveni Ranije opisene rutina OPEN ovo obavlje eutomatski eli ovde mi moramo o tome da vodimo računa Sekundernoj edresi se more posvetiti dodetne pažnja i eko se otvere datoteka ne disku eli sa ovda načemo upuštati u detelje

Nakon upoznavenje sa ovim elemanternim rutineme postaje jsano kako su izvedene ostele Teko ne primar CHKOUT izvodi LISTEN i SECOND uz jeden dodetak eko ete upotrebili sekundernu edresu i kojoj je bit 7 jedinica dekle veču od 127 onde kao tekve naše biti postate već će se apoljnjem uredeju etaviti do znenje de sekundarne edrasa ne sledi što eveki od njih može protumečiti ne svoj nečin Štempač će se ne primer postaviti keo de je primlo sekundarnu adresu 0

Moglo bi se reći de prsanje programe ne ovej nečin ima opravdenje semo u slučeju eletamskih programe Oveko se izbegava evaki moguč nseporazum oko toga de li je korisnik u programu upotrebio logič ' broj veze koji je i nema potraben e i izvršavanje programe je nešto brže

Zoran Životić

Programiranje u bejziku | funkcije, potprogrami, procedura

Kad god je potrebno da se neki postupak više puta ponovi u programu onda je zgodno da se naredbe koje ga opisuju izdvoje u posebnu celinu — potprogram (sabroutine) koja bi se pozivala iz svake tačke programe u kojoj je neophodno primeniti taj postupak i, po izvršenju vracala upravljenje glavnom programu na naredbu koje sledi neposredno iza poziva potprograma Ovakva organizacija programe omogucava njegov kraći zapis Ako je zapis postupke zahteveo n programskih redove i ponavljao se za razne argumente m puta u programu onda umesto n*m programskih redova bez korišćenja potprograma pišemo samo n redova (+m poziva i ber m povratka u program) Ovo ima ze posledicu smenjenje verovatnoce grešaka pri pisanju programe olakšano testiranje i uštedu memorijelnog prostore All sve ima svoju cenu — gubi se na brzini izvršavanja programa Dužina izvrševenje programe uvecava se za vreme potrebno ze realizaciju naredbi poziva i povratake i prenos argumenata Potpuno sprovedene koncepcija potprograme podrezumeva da su oni potpuno izdvojene celine nezevisne od glavnog programe te se preme tome mogu koristiti iz raznih programa i imeju sopstvene — lokalne promenljive koje nemeju nikakve veze sa istoimenim promenljivim glavnih programa To omogućave timski rad ne pisanju aplikacija jer je dovoljno da jedan programer da globalni opis rešenje e razradu detalja može prepustiti svojim kolegama koji će to učiniti u zasebnim potprogramima Glavni programer projektuje samo veze izmedu ovih celina i naznačava koje su promenljive globalne tj zajedničke za program i poje dine potprograme

Uvedene funkcije

Odmeh de kažemo de bejzici računera o kojima pišemo nemeju ovu poslednju mogucnost dakle imaju vrlo veliku manu (napominjem da ovde ne govorimo o sajmonsu i drugim proširenim bejzicima koji imaju tako dobre i bogate mogucnosti da zeslužuju da se pomenu u svakoj teorlji programskih jezika) Ono čime raspolazemo u bejzicima kucnih računara su mogucnost uvodenja funkcije tzv funkcijskih naredbi (sa izuzetkom galaksije ali i ZX81 TRS 80 i drugih koji ih nemeju) i korišćenje internih potprograma — potprogramskih segmenata koji mogu de stoje jedino u okviru glavnog programe Nesto bolje mogunosti imaju jedino Acornovi racuneri

Računari obično imaju ugradene potprograme za računenje elementarnih funkcija koji se pozivaju navodenjem imene funkcije se konkretnim argumentima ze koje treba izračunati vrednost Naredba

```
PRINT SIN (0)
```

poziva ugradeni potprogram za računenje vrednosti funkcije sinus po manje ili više dobroizabranoj približnoj metodi Argument 0 je ulazna veličine u potprogram, stvarni argument za koji se računa vrednost po postupku opisanom ze fiktivne argumente Izračuneta vrednost se zetim vraća u program (ovde u naredbu u direktnom rezimu) i štampa Nemaju svi računeri isti broj ugradenih funkcija niti je računanje njihovih vrednosti istog kvaliteta Od računare o kojima govoriomo najlošija je situacija se galaksijom — one ne samo što ime najmanje funkcija nego su one i najlošije uradene (kao uostalom i ne TRS 80) Nista bolje situecije nije ni kod „spektruma — to sigurno nije njegov adut „Komodor 64 ima podnošljivo uradene elementarne funkcije sa ponekim „biserom (o tome ćemo drugom prilikom) e Acornovi računari mada nisu oslobodeni bagova nude pristojne mogućnosti

Mogućnoat uvodenje korieničkih funkcija je veoma vežne ne samo što teko možemo definisati sinus hiperbolički neki obrazac za proreču kamate ili bilo koju drugu funkciju koja nam je potrebna već i zato što možemo umesto loše isprogramirane ugradene funkcije da uvedemo svoju koja ce davati tačnije vrednosti Ne želost kao što rekosmo ranije galaksijin bejzik nem ne pruža ovu mogucnost

Uvedene funkcije — funkcijske naredbe delinišu se posebnom naredbom oblike

```
DEF   <ime funkcije>  (<argumenti>)=<izraz>
```

ime uvedene funkcije obavezno počinje slovime FN za kojime slede jedno ili dva slova kod komodora (drugi simbol može biti i cifra a teorelski je moguce da se stavi i duze ime ali kao takvo nece biti upamceno pa to praktično nema smisla — dakle neslavak moze biti ime realne promenljive ime azbučne ili celobrojne promenljive uzrokuje prijavu greške) a kod spektruma i Acorna imamo vecu komociju — ime moze imeti i više simbola

Po broju argumenata koji se mogu naveeti opet „komodor ime lošije mogućnosti od druga dve racunaia — može da prihvati samo jedan argument ze razliku od više njih koliko je dozvoljeno za spektrum i Acorn i što je naročito velik hendikep za komodore — on omogućava jedino uvodenje numeričkih funkcija dok druge dve računara mogu da rade i sa uvedenim azbučnim funkcijama Ilustrujmo korišćenje funkcijske naredbe jednim jednostevnim programom

spektrum

```
10 CLS
20 PRINT  BROJ    KUB
30 FOR a  1 TO 20
40 PRINT a  FNk(a)
50 NEXT a
60 DEF FNk(x)=x*x*x
```

komodor

```
10 DEF FNK(X)=X*X*C
20 PRINT BROJ , „KUB
30 FOR A=1 TO 13
40 PRINT A  FNK(A)
50 NEXT A
```

Acorn

```
10 CLS
20 PRINT , BROJ , „KUB
30 FOR A=1 TO 13
40 PRINT A  FNK(A)
50 NEXT A
60 END
70 DEF FNK(X)
80=X*X*X
```

Možde vam se čini de je ovo programe bez smisle jer treći stepen možele dobiti i pišući jednostavno X↑3 ali grešite jer se oveko računa tačnije i skoro deset puta brže vrednost treceg stepena U to se možete uveriti ako izbacite definiciju funkcijske naredbe i umesto FNK štampate vrednost A↑3 (Radi boljeg uvide u brzinu stavile de ciklus ide do 1000)

Snaga argumenata

Ono što ste verovetno primetili iz ova tri listinge je de se kod Acorna ne mora definisati vrednost u jednom programskom redu Po tome kod njega uvedene funkcije u stveri prevezilaze konvencionalni pojam funkcijske naredbe jer omogućavaju da se postupak za računenje funkcije definiše u više programskih redova — dakle pre bi sa mogło reći da je ovo analogon ze funkcijski potprogram u fortranu (koji, za razliku od opšteg potprograma ime semo jednu izlaznu veličinu) Kada se još dode mogućnost rekurzivne definicije funkcije o kojoj ćemo posebno govoriti dobijaju se moćna svojstve o kojime možde niste ni slutili Formalna razlika u odnosu ne definisanje funkcija kod drugih računere je da se opisna naredbe (poput DATA na primer) DEF FN mora nelaziti ize fizičkog kraja programe „Spektrum , pak, uvak traži DEF FN od početka programa ked se pomene ime neke uvedene funkcije pe je naročito kod dužih programa dobro sva funckije definisati na samom početku programa

Argumenti koji se navode izmedu zagre de u definiciji funkcijske naredbe zovu se fiktivni argumenti Fiktivni jer se raztikuju od promenljivih u programu — za njih program ne razerviše posebne memorijske registre i možete sasvim nezevisno od fiktivnih argumeneta u programu koristiti istoimene promenljive Možemo fiktivne argumente smetrati nekom vrstom lokelnih promenljivih za funkcijsku naredbu

Rekli smo da „spektrum i Acorn računeri mogu de koriste uvedene funkcije sa više agrumenata Otkucajte NEW i unesite sledeći program

Danas je nezamislivo da se rešenja bilo koja veće aplikacije daje u obliku jednog programa. Koncept potprograma, predložen u samu zoru razvoja programskih jezika, zahvaljujući nesumnjivim olakšicama koje pruža, još nije išao e top liste omiljenih programerskih tehnike. Ove koncepcije ugrađene je, sa više ili manje mogućnosti, u ave više programska jezika. Da bismo analizirali koliko je uspešno izvršeno njeno ugradivanje u bejzike kućnih računara, zadržimo se prethodno na samoj ideji potprograme i pokušajmo da sagledamo šta ave ona nudi.

„Spektrum", Acorn

```
10 DEF FNa(a  b  c  d  f)=a b*b
20 PRINT  unesite dva broja
30 INPUT a  b
40 PRINT FNa(a  b)
```

Posle startovanja programa videcete da on — ne radi. Ali ako promenite liniju 40 na sledeci način

```
40 PRINT FNa(a  b, O  O, O)
```

program ce raditi. Upamtite da se prilikom svakog poziva funkcije moraju navesti stvarni umesto svakog fiktivnog argumenta.

Potprogramski segmenti

Mnogo više mogućnosti na svim računarima pružaju potprogramski segmenti, posebni delovi programa na koje se upravljanje prenosi naredbom GOSUB <broj prog. reda> (CALL kod „galaksija"). Postoji znacajna razlika izmedu naredbi GOTO i GOSUB. Realizacija GOTO naredbe podrazumeva da se u mesto gde se čuva adresa sledece naredbe slavi adresa naredbe na koju ukazuje GOTO. GOSUB pak uz sve ono što proizvodi GOTO još i čuva adresu naredbe koja je bila zapisene neposredno iza nje (upisuje je u stak). Ovo omogucava da se po izvršavanju potprogramskog sagmenta osveži secanja na masto odakle je pozvan potprogram i vrati mu se upravljenje. Taj povratak u program ostvaruje se naredbom RETURN (to nije tipka RETURN kod komodora) koja vraca upravljanja ne poslednju adresu upisanu u stak.

Zahvaljujuci staku moguća je vršiti i pozivanje potprograma po dubini. To znači da glavni program zova potprogram 1, potprogram 1 zove potprogram 2 i tako redom a povratak sa vrši obrnutim redom od reda pozivanja. Specijalan slučaj pozivanja potprograma po dubini je rekurzivno pozivanje potprograma — slučaj kada potprogram poziva samog sebe. Na školskom primeru računanja vrednosti funcije n! pokazaćemo kako izgleda i kako se realizuje rekurzivno pozivanja potprograma u bejziku koji (sam kod funkcijske naredbe za Acron) nije projektovan za rekurziju i na osnovu najveće vrednosti n za koja se moze izvršiti izvesti zaključka o lome do koje dubine se mogu pozivati potprogrami na svakom od računara.

„Spaktrum", Acorn, „komodor"

```
1 REM GLAVNI PROGRAM
10 I  O
20 N  I
30 F  1
40 PRINT N  !
50 GOSUB 100
60 PRINT F
70 I=I+1
80 GOTO20
90 STOP
99 REM POTPROGRAM
100 IF N=O THEN RETURN
110 F  F N
120 N=N  1
130 GOTO 100
```

(napomena za spektrum: treba kod naredbi dodele staviti službenu reč LET)

Startujte ovaj program i ulvrdićete da sa za svaki računar on završava porukom o prekoračenju posle izračunalog 33!

Izmenite sada potprogram na sledeći nacin

```
99 REM REKURZIVNI POTPROGRAM
100 IF N=O THEN RETURN
110 N  N-1
120 GOSUB 100
130 N=N+1
140 F  F N
150 RETURN
```

„Galaksija"

```
1  GLAVNI PROGRAM
10 I=O
20 N  I
30 F  1
40 PRINT N  !
50 CALL 100
60 PRINT F
70 I  I+1
80 GOTO 20
90 STOP
99 ! POTPROGRAM
100 IF N  0 RETURN
110 F=F*N
120 N=N-1
130 GOTO 100
```

Analizirajmo kako ce se izvrsavati ova potprogram kada se pozove za N=1. GOSUB u liniji 30 prenosi upravljanje na liniju 100 a edrasa 40 ide na stak. Kako N nije O ide se na liniju 110. Seda N postaje O pralazi se na naredbu 120 u kojoj potprogram zove samog sebe. To znači da u stek iznad 40 upisuja adresu 130 i pralazi na liniju 100. Pošlo je zadovoljen uslov vrsi se povratak na adrasu koja se uzima iz steka tj. 130. N se uvaćave za jedan. F se mnoz sa N i slize se do linija 150 koja opet vrs povratek u program ali sada na adresu 40 jer ja po uzimanju 130 ona ostala na vrhu sleka.

Na ovaj način dobićamo kao u prethodnom slučaju tablicu vrednosti faktorijela ali ne zaključno sa 33 (sem kod spektruma koji koristi mašinski slek). Program se na „galaksiji" završava vec kod N=13 a na komodoru i Acornu imamo nesto veci stek predvidan za GOSUB pa dobija-

➔

vrednosti do 23. Sada je razlog za prekid programa ? OUT OF MEMORY ERROR što na znači da nama uopšte slobodnog memorijalskog prostora, vec da ga nama tamo gde je potreban — na staku

Kule Hanoja

Ovaj primer nije izabran da demonstrira prednosti rekurzija (bila bi to antipropaganda) već da pokuša da objasni kako se u stvari realizuju instrukcija GOSUB i RETURN i odmah koliko se duboko mogu pozivati potprogrami. Pravu rakurziju ćemo ilustrovati na primeru korišćenja funkcijska naredbe Acornovog bajzika

Pošto smo smatrali da je funkcija n! svima poznata (proizvod broja n sa svim prirodnim brojavima koji mu prethoda i 0!=1) nismo je navodili, ali za ovaj drugi primer navodimo prirodnu rakurzivnu definiciju funkcija n!

N!=1 za N=0
N!=N*<N-1>! za N>0

Pokušajmo da dokažamo da je 3! isto što i 1*2*3. Po gornjoj definiciji je 3!=3*2! Dalja je 2!=2*1!, a 1!=1*0!. 0! je 1 po definiciji pa je onda 1!=1*1=1 onda je 2!=2*1!=2*1=2 a 3!=3*2!=3*2=6. Slično tome bismo mogli da pokažemo da je 5!=120 <pokušajte!> i da na taj način bar poverujamo u gornju definiciju faktorijala ako ne i da matematički dokažemo njenu ekvivalentnost sa ranije izloženom. Sastavimo sada novu varziju ranija funkcija za faktorijal (slika 5)

```
slika 5:

3000 DEFFNfaktorijel(N)
3010 IF N=0 THEN =1 ELSE =N*FNfaktorijel(N-1)
```

Na prvi pogled bi se rakio da je ovako dobijana funcija mnogo bolja od ona koju smo napisali pre nego što amo znali za rekurzija: dva linija umesto sedam i nijedna pomoćna promantijiva (ranija smo koristili dve S i I) Moramo madutim da kažamo da ja ova funkcija veći potrošač memorija. Za N=25 će rakurzivno biti pozvano 25 nivoa potprograma ako se za pozivanja svakog nivoa troši po osam bajtove za belažanje adresa povratka i drugih informacija, plus još bar 6 bajtova za amaštanje novog fiktivnog argumanta (uvek sa zove N), dobijamo da bi sa primanom ove funkcija, uz tridesetak kitobajta RAM-a, mogio najviše izračunati 2000! (2000! sa, jasno, na može izračunati, jer ja opsag brojava sa kojima berata računar ograničan,dok ja 2000! broj sa prako 5730 cifara), dok u prvom primaru, kada amo koriatili patju, takvog ograničenja nija bilo. Uopšta važi da su programi koji koriata rekurzija naoko kratki, alagantni i jednostavni, all su veliki potrošači memorija i računarskog vramana, pa Ih traba izbegavati kad god je to moguće (a moguća je uvek, uz više Ili manja problama)

Završavajući ovu kretku priču o rakurzijama, ne možemo da na navedamo jadan divan primar njihova upotrebe. Varovetno vam ja poznat problam pod nazivom „Kula Hanoja"! priča sa da u gredu Banaresu postoji hram u koma ja induski bog Brama pri stvaranju sveta postavio tri dijamantska štapića i na jedan od njih stavio 64 zlatne kotura različitih veličina na zamlju ja stevio najveći kotur, na njaga manji i tako sva do najmanjeg kotura koji sa nalazi na vrhu ovako dobijena kupa. Stotinu svaštanika toga hrama bez prestanka danju i nocu pranosi koturova sa prvog na drugi atubac služaći se trećim kao pomoćnim i poštujući ograničanja koja ja zadao Brama: sma sa premaštati samo jadan po jedan prstan i nikada sa veći prstan na sma staviti prako manjeg. Kada svaštanici budu obavili ovaj na prvi pogled jednostavan posao nsatupiće kraj svata (na piašita sa jednostavan račun pokazuja da nam ja do kraja sveta ostalo još nakih 500 milijardi godina). Sastaviti program koji rašava ovaj problam bez rakurzija nija ni malo lako sastaviti program sa rakurzijama pradstavlja pravu dačju igru — pogledajta sliku 6 i 7 na kojima ja osim programa pisanog na BBC bajziku, data i varzlja na paskalu koju će varovatno isprobavati vlasnici „spektruma i komodora

Opšti potprogrami

Konačno zadržimo sa na mogućnosti korišćenja procedura koju imaju jedino Acornovi računari

Sva funkcijska naredba i opšti potprogrami sa stavljaju iza glavnog programa i, osim što moraju da imaju linijska brojave koji su veći od linijskih brojava naredbi glavnog programa, predstavljaju potpuno nazavisna celina koja imaju svoja ulazna i izlazna tačka. ulazna i izlazna veličina i

slika 6:

```
 10 REM
 20 REM
 30 REM          Kule Henoje
 40 REM
 50 REM     Dejan Ristanovic 1
 60 REM
 70 REM
 80 REM          Racuneri 7
 90 REM
100 REM
110 INPUT "Koliko diskova";N
120 CLS
130 PRINT "DISK","SA:", NA:"
140 PROCtowers(N,1,2,3)
150 END
200 DEFPROCtowars(N,X,Y,Z)
210 IF N=0 THEN ENDPROC
220 PROCtowars(N-1,X,Z,Y)
230 PRINT ;N,;X,;Y
240 PROCtowers(N-1,Z,Y,X)
250 ENDPROC
```

lokaina promenjlive. Posmatrajmö naredbe sa silka 1

```
slika 1:

1000 DEFPROCispis(AX,BX,X,Y)
1010 LOCAL I,J
1020 FOR I=1 TO AX
1030     FOR J=1 TO BX
1040         REM  radni deo petlje
1200     NEXT J
1210 NEXT I
1220 ENDPROC
```

```
slika 7:


ISO-Pascal compiler V. R1.00

   1 0 - program hanoj(input,output);
   2 0 - (*  rakurzivno resenje problema 'Kule Henoje'
   3 0 C                Dejan Ristanovic 1985              *)
   4 0 -
   5 0 - const maxn=64;
   6 0 -
   7 0 - var n:integer;
   8 0 -
   9 0 - procedure prenos(N,X,Y,Z:integer);
  10 1 - begin
  11 1 -    if N<>0 then begin
  12 1 -                      prenos(N-1,X,Z,Y);
  13 1 -                      writeln('Se',X:3,' na',Y:3);
  14 1 -                    prenos(N-1,Z,Y,X)
  15 1 -                  end
  16 1 - end;
  17 1 -
  18 0 - begin   (* glevni program *)
  19 0 -    write('?');readln(N);
  20 0 -    if n>maxn then writeln('Cekej do kraja sveta')
  21 0 -              else prenos(N,1,2,3)
  22 0 - end.
  23 0 -
  24 0 -
0 Compilation error(s)
Code size = 220 bytes
```

Ovim kratkim segmentom je opisana jedna procedura. Njeno ime je **lepša** i ima četiri takozvana fiktivna argumenta A%, B%, X i Y, od kojih su prva dva celi, a druga dva racionalni brojevi. Zašto ova promenljive nazivamo fiktivnim argumentima? Da to razumemo, pogledaćemo sliku 2, primer segmenta koji poziva proceduru koju smo upravo definisali.

Primetili smo da je u poslednje vreme među našim čitaocima sve više vlasnika „amstrada". Shodno tome od sledećeg broja, računara u školi bejzika uključiti i bejzik „amstrada dajući na početku pregled mogućnosti i nedostataka ove mašine u poređenju sa ostalim mašinama koje smo do sada obrađivali u dosadašnjim brojevima.

```
slika 2:

1[illegible] A%=5:B%=6
110 INPUT "Unesite x i y koordinatu tacke";X,Y
120 PROCedura(A%,B%,X,Y)
130 PRINT A%,B%
140 END
```

U prvoj liniji ovog programa promenljiva A% dobija vrednost 256, a zatim se sa tastature unose dva cela broja koja „pamte" promenljive DONJA% i GORNJA%. Sledi kritična linija, poziv procedure. Bejzik interpretator će najpre da pronađe proceduru i uporedi argumente

DONJA%
GORNJA%
25
73
stvarni argumenti

A% B% X Y fiktivni argumenti

Vidimo da se fiktivnim promenljivima X i Y dodeljuju vrednosti 25 i 73 respektivno, dok se fiktivnim promenljivima A% i B% dodeljuju vrednosti stvarnih promenljivih DONJA% i GORNJA%. Obratimo međutim pažnju na stvarnu promenljivu A% kojoj je u prvoj liniji programa dodeljena vrednost 256. U potprogramu se, takođe, nalazi promenljiva A% kojoj je, vrlo verovatno, dodeljena neka vrednost različite od 256. Dok se potprogram izvršava, A% će imati tu vrednost. Kada se međutim nađe na naredbu ENDPROC, kontrola će se vratiti glavnom programu, pri čemu će 256 ponovo postati aktualna vrednost promenljive A%, tako da će naredba 130 ispisati baš ovaj broj.

Dodajmo sada našem glavnom programu jednu liniju, tako da glasi kao na slici 3.

Startujmo ovako prepravljeni program i dobićemo ako se izuzme ono što radi **lepša**, rezultate 256, 12 i 14. Očekivano? Na prvi pogled jeste, ali ne smemo da zaboravimo da se u proceduri **lepša** koriste promenljive I i J kao brojači u petlji, što znači da im je vrednost morala biti promenjena. Da to proverimo, ubacićemo u potprogram liniju

1215 PRINT I, J

i dobićemo vrednosti I i J u proceduri koje su jasno različite od 12 i 14. Obratimo međutim pažnju na naredbu **1010 LOCAL I, J** kojom su promenljive I i J deklarisane kao lokalne za tu proceduru. Slično formalnim argumentima, lokalne promenljive imaju vrednost samo u proceduri i po povratku u glavni program postaju nedefinisane ili dobijaju vrednosti koje su imale pre poziva potprograma (da je u proceduri pisalo LOCAL I, J, K a zatim K=100 u glavnom programu bi linija 140 PRINT K dala poruku o greški: No such variable.)

Suprotnost lokalnim promenljivima su takozvane globalne promenljive. Ako u glavnom programu dodamo liniju 5 M=123 a u listi fiktivnih argumenata i u listi lokalnih promenljivih ne pominjemo M, PRINT M će kako u programu tako i u

```
slika 3:

100 A%=256
105 [illegible] J=14
110 INPUT "Unesite [illegible]
[illegible]
120 PROCedura(A%,B%,X,Y)
130 PRINT A%,B%
140 END
```

potprogramu dati broj 123. Ukoliko slično tome u proceduri promenimo vrednost promenljive M, te će promena biti u važnosti i posle njenog završetka. Standardni bejzici imaju isključivo globalne promenljive i jedini jezikom predviđen način za povezivanje potprograma je naredba GOSUB (CALL).

Dejan Ristanović
Nevenka Spalević

mali oglasi

SPEKTRUM

NEW DATA

SPECTRUM — AMSTRAD — SCHNEIDER
Programski paketi za stručnjake
STATISET
LINSET
GEODET
ANNUITY
Posebne za Amstrad
CHECKER — program za testiranje tastature joystick-a i kasetofona! Uverite se da je sve o.k.
NEW DATA, O. Brašovane 8/10, 21000 Novi Sad

VAŠNO

Spektrumevci pronađite svoj žbor od interesantnih zemljaka najnovijih programe za Spectrum 48 — najnovije hitove programe video giga memorije vaše ili neke kasete povoljno po prištupačnim cenama. Literatura besplatni katalog. **Bejić Goran**, Slevena Filipovića 29/85, 11040 Beograd, tel. 011/635-205.

SPEKTRUM prodajem komplet od 12 čipova za povećanje memorije Spektruma se 16 na 48Kb sa uputstvom za ugradnju. Cena 14 000 dinara. Po želji vršim ugrađivanje (1 000 din.). Svoj Spektrum možete poveceti de me i 80 Kb. Tekav komplet čipova staje 20 000 d. nare. Takođe dobijate uputstvo za ugradnju, prekidač za preklapanje memorije. Za Spektrum prodajem i Kempston interfejs za dve džojstika — (cena 10 000 d.) i nove džojstike Big Shot (5500 d.)
Ignjatović Đorđelev
Lole Ribara 1/17
16000 NIŠ

• Najjeftiniji Spectrum programi od 7 d. nare nađete zbor preko 300 odličnih programa. **Đevinovski Saša**, Gajeva 4, 43400 Virovitica.

KOMODOR

• Prodajem 6 portne konektore za kasetofonski port za Commodore 64. Vladimir Ilić, Đ. Kidriča 5, 22300 Stara Pazova, tel. 022 311 013.

COMMODORE 64 — Najbolji programi za kasete: SUMMER GAMES, IMPOSSIBLE MIS, RAID OVER M, HAVOC, JET SET WILLY, PYJAMARAMA i ostali. Za diskete: CONAN, BROAD STREET, GI JOE, HOBBIT 2, MERLIN assembler i mnoge druge. Spisak besplatan. DENI OZREN **Đukić**, 41020 Zagreb, Čelogovićeve 5 III, tel. 041 686-004.

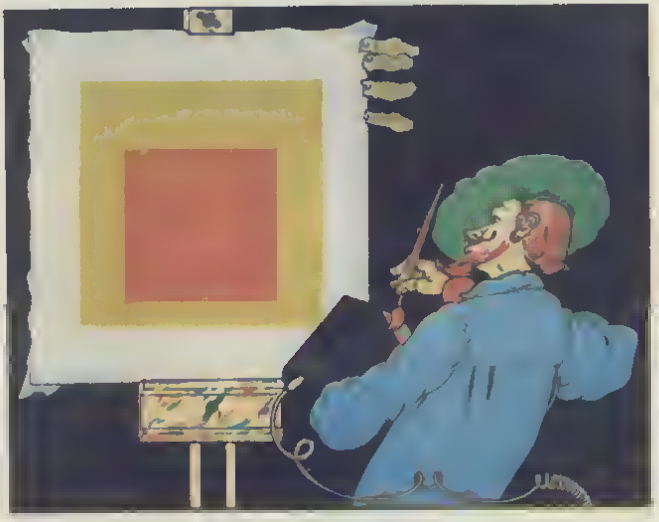
KOMODOR 64 — prodajem originalnu ležišnu servisnu shemu (formata A3 × 2). Cena sa PTT troškovima — 400 din. pouzećem. Za Komodor 64 takođe prodajem CP/M modul. Kada se priključite u user port postaće Vam dostupan mnogi CP/M program. Cena 11 000 dinara. Prodajem i nov specijalni kasetofon za Komodor (14 000 d.) i nove džojstike Big Shot (5 500 d.) TDK kasete sa 25 korisničkih programa i 25 najboljih igara — samo 3 000 din. pouzećem.
Ignjatović Đorđelev
Lole Ribara 1/17
1500 NIŠ

• Za commodore 64 više od 1700 programa! Niske cene! Seveti za početnike! Specijalne pogodnosti i popusti! Razmena! Tražite besplatan katalog! **Mirko Žegarac**, Vukasovićeva 92, 11080 Zemun, tel. 011/592-024.

RAZNO

• Pored adaptera koji svakom kasetofonu omogućavaju rad sa CBM-64 ovde ćete naći štiklu izbor programa. Cena — prava sitnica. Vladimir Ilić, Đ. Kidriča 5, 22300 Stara Pazova, tel. 022/311-013.

Grafika na računaru — kako to radi „amstrad"



Po svojim grafičkim mogućnostima (kao zbog ostalih izvanrednih karakteristika) CPC 464 se među kućnim kompjuterima nalazi pri samom vrhu. Poslo tih kompjutera kod nas ima sve više logično ga je uporedivati s ostalim najpopularnijim računalima — spektrumom commodorom 64 i BBC-em B.

Pri radu s spectrumom sloji na raspolaganju upotreba teksta formata od 32 znaka u 24 reda te grafika visoke rezolucije od 256x192 točke. Na raspolaganju ima 8 boja. Definiranje sprajtova nije sadržano u spectrumovim osnovnim mogućnostima ali se može naknadno programski ostvariti.

Komodor 64 ima prikaz teksta od po 40 znakova u 25 redova te grafiku visoke rezolucije 320x200 točaka. Moguće je koristiti 16 boja te definirati 8 sprajtova. Mogućnosti računala C64 su bolje nego spectrumove. Upotrebu grafike otežava korištenje instrukcija PEEK i POKE, odnosno potreba za direktnim pristupom određenim memorijskim lokacijama.

BBC B ima znatno bolje grafičke mogućnosti i od spektruma i od C64 a nešto malo bolje i od CPC-a 464. Maksimalni format teksta je 80 znakova u 32 reda. Ovisno o modovima dopušta se do maksimalne grafičke rezolucije od 640x256 točaka što je neznatno bolje od CPC-a ali je izbor modova veći. BBC B ima svega 8 boja a sprajtovi se moraju naknadno programski definirati. U biti moguće je korištenje 16 nijansi.

Amstrad/Šnajder ima maksimalni format teksta od 80 znakova u 25 linija: tri moda grafike, te najvišu moguću rezoluciju od 640x200 točaka. Moguće je koristiti 16 od 27 boja a sprajtovi se moraju posebno definirati. Iz svega ovoga se vidi da uz odnos cijena kvalitetu CPC 464 za one koji žele da crtaju predstavlja najbolji izbor.

Grafički modovi i upotreba boje

Kod ve[illegible] broja računala izbor grafičkih modova[illegible] s o željenom broju boja koji se želi upotrijebiti za prikaz. Točnije to znači da rezolucija slike ovisi o broju upotrijebljenih boja — veća finoća povlači za sobom upotrebu manje boja i obratno. Ze ove činjenice je CPC 464 dobar primjer.

Tri moda rada u kojima CPC 464 može raditi su označeni kao modovi 0 1 i 2.

Mod 0 daje najgrubiju grafiku ali se zato u tom modu može upotrijebiti 16 boja. Tekst je podijeljen na 20 znakova u 25 linija. Grafičke rezolucija je 160x200 točaka (160 u liniji i 200 vertikalno). Taj mod se najčešće upotrebljava u igrama kod kojih se želi upotrijebiti što više boja a kvaliteta prikaza nije od osobite važnosti.

Mod 1 je osnovni mod a to znači da se računalo nalazi u njemu odmah po uključenju. Tekst je podijeljen na 40 znakova po retku a grafička rezolucija je 320x200. Moguće je koristiti samo 4 boje odjednom. Taj mod je vrlo dobar za normalno programiranje za poslovnu i zananstvenu grafiku gdje je potrebno prikazati više detalja na ustrb korištenih boja.

U modu 2 tekst se prikazuje sa 80 znakova po liniji. Grafička rezolucija je najveća u ovom modu (640 – 200 točaka). Mogu se koristiti samo 2 boje. Taj mod se upotrebljava tamo gdje je potrebna vrlo fina grafika te za obradu teksta i tabela (word-processing i spreadsheets). Ponekad se javljaju problemi pri prikazu 80 znakova u [illegible] zbog nestabilne slike na monitoru.

CPC 464 može proizvesti 27 različitih [illegible]a. Dakle kao da imamo 27 patrona s [illegible]bojom i tim odnosno 2 pera (PEN) za [illegible]anje u[illegible]zuetno viskoj gralici 4 za [illegible]ma[illegible]ne crteže i 16 za grube crteže. Može se dakle koristiti boja tinte (INK) prema želji samo mora biti različita od boje pozadine.

PAPER INK i PEN su instrukcije sadržane u CPC ovom Locomotive bejziku. INK predstavlja jednu od 27 upotrebljivih boja. PEN je odabiranje željenog pera za crtanje a PEPER-om se odabire boja pozadine. U praksi sva pera imaju različite vrijednosti pa često nije potrebno koristiti instrukciju INK odnosno moguće je koristiti boju naznačenu od kompjutera.

Ako se na primer utipka PEN 2 i pritisne ENTER — poruke READY će biti obojena u svijetlo plavo.

Pere moda 1 jesu plava (PEN 0) žuta (PEN 1) svijetlo plava (PEN 2) i crvena (PEN 3). PEN 0 se koristi za pozadinu (PAPER) a PEN 1 za bojanje prednjeg plana. Kada se utipka PAPER 1 tekst će biti u crvenoj boji na žutoj pozadini. Jasno je da u slučaju iste boje teksta i pozadina tekst postaje nevidljiv.

Uz instrukciju INK dolaze dvije vrijednosti npr. INK 1 2. Prvi broj uz instrukciju je broj pera koje želimo koristiti a drugi broj je oznaka upotrebljene tinte odnosno boje (0—26).

Moguće je označiti svih 16 pera ali treba znati da broj pera ovisi o modu u kojem radimo. Tako za mod 0 imamo pera 0—16 za mod 1 pera 0—3 i u modu 2 pera 0 i 1. Odabrano pero nije ovisno o promjeni moda tj. ako odaberemo pero 3 u modu 1 (crveno) te potom mod promijenimo u mod 0 tekst će ostati u crvenoj boji.

U slučaju da se želi odebrali pero veceg broja nego što je moguće u modu u kojem se trenutačno radi promijeniti će se vrijednost npr. pero 4 u modu 1 će dati pero 0 u višem modu odnosno pero (pen) 5 će biti pero 1. Ako se odabere pero 6 u modu 1 dobili će se pero 2 u višem modu.

Kućni kompjuteri ava više privlače korisnika i svojim grafičkim mogućnostima. Uzevši kao primjer više nego praktičan kompjuter amstrad/šnajder CPC 464, može se tvrditi da kompjuterska grafika ne mora biti samo u funkciji igara i ilustracija proračuna, već i soliden poligon za kreativno stvaralaštvo.



Računar izvanrednih grafičkih potencijala Amstrad/šnajder CPC 464

Demonstracija boja

Program ‚Demonstracija boja' prikazu je kolor mogucnosti moda 0 Prva petlja linije 35 do 65 boje linije teksta u svaku od 16 mogucih boja Linija pen 0 se ne može uočiti jer je iste boje kao i pozadina Drugi dio programa linije 75 do 125 čine petlju FOR NEXT koja omogucuje da svih 16 pera kružl kroz sve moguće boje Ostatak programa resetira čelirı pera koristeci mod 1

```
  5 REM demonstracija boja
 15 MODE 0
 25 CLS
 35 FOR boje=0 to 15
 45 pen boja
 55 PRINT "To je pen " boje
 65 NEXT boja
 75 FOR a=0 TO 15
 85 FOR boja=0 TO 26
 95 INK a, boje
105 FOR k=1 TO 200 : NEXT k
115 NEXT boje
125 NEXT a
135 MODE 1
145 INK 0,1
155 INK 1,24
165 INK 2,2
175 INK 3,3
185 PEN 1
195 END
```

Program se može modificirati lako da resetira svih 16 pera (penova) koristeci FOR NEXT petlju te instrukciju DATA Isto tako mogu se linije 125 do 195 zamijeniti sa CALL &BBFF, tj subrutinom operativnog sistema koji resetira boje ekrana modove ali ne i postojeci odabir pera (penova)

Instrukcija INK moze takoder postaviti pero tako da se izvodi mesing efekt, odnosno bljeskanje. To se postiže određivanjem dvije boje (dva INK-a) koji ce se alternativno koristiti Fleširanje je najjednostavnija poslici postavljenjem boje INK-a (tinte) na neku različitu pozadinu Iako da se u jednom momentu fleširajuci objekt razlikuje od pozadine a u drugom momentu stapa s pozadinom

Brzina fleširanja — bljeskanja se određuje instrukcijom SPEED INK Uz instrukciju dolaze dva broja koji moraju imati cjelobrojne vrijednosti Prvi broj određuje vrijeme trajanja prve boje a drugi vrijeme trajanja druge boje Dakle pomocu ove instrukcije moguce je poslici točno željeno vrijeme izmjene fleširajućih boja

Crtanje linija

CPC 464 raspolaze setom instrukcija za crtanje linija i plotiranje točaka na ekranu Pri tome se koristi sistem kordinata za odredivanje vertikalnih i horizontalnih pozicija

Gratička rezolucija je u sva tri moda različita ali koordinatne vrijednosti ostaju iste To znači da se kroz dani set DRAW instrukcija može nacrtati lik na istom mjestu u sva tri moda što predstavlja vrlo korisnu mogucnost U stvari sve tri moda imaju istu vertikalnu rezoluciju — 200 linija Koordinatni sistem koristi vertikalnu skalu do vrijednosti 400

To znači da koordinate 0 i 1 odgovaraju jednoj stvarnoj točki 2 i 3 drugoj itd Horizontalne koordinate idu do vrijednosti 640 To odgovara najvećoj rezoluciji tj modu 2 Mod 1 ima 320 točaka u liniji dakle koordinate 0 i 1 su u stvari predstavljene kao jedna stvarna točka 2 i 3 kao druga itd kao u slučaju kod vertikalnih koordinata Kod moda 0 koordinate 0 i 2 i 3 predstavljaju jednu točku 4 5 6 i 7 drugu itd

Grafičke koordinate imaju vrijednosti od 0 do 639 horizontalno te 0 do 399 vertikalno Početna lokacija (koordinate 0,0) koja se nalazi u donjem lijevom uglu naziva se ‚origin To je različito od pojma ‚text-origin koji se nalazi u gornjem lijevom uglu Posebna pogodnost je ta što se ‚origin može premještati bilo kude po ekranu koristeci instrukciju ORIGIN

Kompjuterom CPC 464 se može crtati i ‚izvan ekrana Takozvane ‚off-screen pozicije ce biti ispravno zabilježene, što u praksi znači de će ako crtamo ‚izvan ekrana, pa se opet vraćamo u njegove okvire, sve biti na svome mjestu, odnosno linije iscrtane pod ispravnim kutevima

Kao što postoji kursor pn tekstualnom prikazu tako postoji i ‚nevidljivi grafički kursor Tako kada se npr iscrteva linija one se crta od početne pozicije kursora prema novo odredenoj poziciji

Postoje dva načine odredivanja koordinata — relativni i apsolutni Apsolutno odredivanje podrazumijeva kako to i naziv govori apsolutno odredivanje na zamišljenoj koordinatnoj mreži ekrana, a može se shvatiti i kao relativno prema ‚oridžinu

Relativno odredivanje predstavlja odredivanje kao relativno pomicanje pozicije kursora Koordinate s negativnim vrijednostima se mogu koristiti pri oba načina odredivanja Karakteristika apsolutnog odredivanja je jednostavnost pa se i više koristi a relativnog — da u praksi pojednostavljuje mnoge crteže

Instrukcije DRAW PLOT i MOVE pred-

➔

slavljeju glavne naredbe amstrad/šnajderove grafike U ovim oblicima se koriste pri apsolutnom odredivanju Za relativno odredivanje imaju oblike DRAWR PLOTR i MOVER

DRAW crta linije od pozicije grafičkog kursora prema definiranoj novoj poziciji MOVE pomiče grafički kursor bez posebnog označavanja na ekranu PLOT je slične instrukcija kao MOVE ali crta točke na novoj poziciji kursora

Sljedeći primjer ilustrira primjenu instrukcija MOVE i DRAW te relativne i apsolutne koordinate Program crta kvadrat na ekranu čija se veličina i pozicija unosi linijama 55 i 65 Dio programa za crtanje kvadrata je pisan kao sabrutina koju čitaoci mogu ugraditi u vlastite programe Linije 115 pokrace grafički kursor na apsolutnu poziciju u liniji 45 Slijedeca četiri linije crtaju kvadrat koristeći ,relativne koordinate Redoslijed iscrtavanje je lijeva strane vrh kvadrata desna strana dno kvadrata Pažnju treba obratiti na činjenicu da crtanja gore i udesno ima pozitivne vrijednosti a dolje i udesno negativne vrijednosti

```
 15 REM crtanje kvadrate
 25 MODE 1
 35 WHILE 1
 45 INPUT  Pozicije donjag ugla
      (x,y)  x,y
 55 INPUT  Visina   v
 65 INPUT  Sirina   s
 75 CLS
 85 GOSUB 105
 95 WEND
105 REM subrutina za crtanje kvadrata
115 MOVE x y
125 DRAWR 0 v
135 DRAWR s 0
145 DRAWR 0 -v
155 DRAWR -s 0
165 RETURN
```

Pri crtanju često trabe izvesti kružnice Ze razliku od npr spectrume amstrad snajderov Locomotiva bajzik nema mo gucnost korištenja instrukcije CIRCLE Me dutim moguće je lako crtati kružnice kori steci funkcije sinus i kosinus (SIN i COS)

Primjer za lakvo crtanje se moze vidjeti u programu

```
  15 MODE 1
  25 BORDER 2
  35 REM crtanje kvadrate
  45 DRAW 0,399,1
  55 DRAW 639,399,1
  65 DRAW 639,0,1
  75 DRAW 0,0,1
  85 REM crtanje kružnice
  95 a=320   b=200   c=196
 105 MOVE a+c,b
 115 FOR d=0 TO 2*PI STEP 0.03
 125 DRAW a+c*COS(d),b+c*SIN(d),2
 135 NEXT d
 145 REM crtanje elipse
 155 c=92
 165 FOR m=0 TO PI STEP PI/5
 175 FOR n=0 TO 2*PI STEP 0.03
 185 p=100-COS(n)
 195 ax=a+p*SIN(m)+c*SIN(n+m)
 205 by=b+b*COS(m)+c*COS(n+m)
 215 IF n=0 THEN MOVE ax,by
 225 DRAW ax,by,3
 235 NEXT n
 245 NEXT m
 255 END
```

Program crta kružnicu koristeći alepsu kao šablonu Linija 25 postavlje boju okvira ekrana a moguce je prema želji, izvesti i tleširanja Broj uz instrukciju BORDER — (2) dolazi iz palete od 27 boja Linije 45 do 75 crtaju kvadrat oko ekrane Pažnju treba obratiti na poseban parametar uz DRAW instrukciju Naime boja linije nije kontrolirana definiranjem pen-a već parametrom uz DRAW koji mora biti broj pen-a Ako boja linije nije označana uz DRAW program koristi posljednje definiranu boju Linije 95 do 135 crtaju kružnicu Varijable a i b su koordinate središta kružnice a varijable c je radijus Linije 155 do 245 crtaju alipsu koja se koristi kao šablone

Upotreba instrukcije PLOT je uobičajena kao i kod drugih računala Jednostavno rutinom za plotiranje točaka no slučajno odredanim mjestima i s slučajno odredenim bojama može se prikazati primjer za upotrebu instrukcije PLOT

```
  5 MODE 0
 15 WHILE 1
 25 PLOT RND(1)*640,RND(1)*400,RND(1)*
    15
 35 WEND
```

Grafički efekti

Jednostavne a efektne crteže je moguce dobiti upotrebom linija i spirale prema sljedecem primjeru

```
  5 MODE 0
 15 BORDER 2
 25 PAPER 5
 35 CLS
 45 a=1   b=350
 55 WHILE B>34
 65 FOR m=0 TO 2*PI STEP 0.1
 75 IF a=5 THEN a=6
 85 IF a=14 THEN a=-1
 95 p=b-50
105 IF p<0 THEN p=0
115 MOVE 320+b*COS(m),200+b*
    SIN(m)
125 DRAW 320+p*COS(m-0.5),
    200+p*SIN(m-0.5),a
135 b=b-1
145 IF p<0 THEN p=0
155 a=a+1
165 NEXT m
175 WEND
185 GOTO 165
```

Linije 135 i 115 točnije element b u njima proizvodi spiralu Druga spirala se tvori linijama 125 i 95 Linije se iscrtavaju točkom prve spirale prema točkama druga spirale Varijabla a odreduje boju izmjenom varijabli mogu se dobiti razni drugačiji grafički efekti

Prozori (Windows)

Kod računala CPC 464 je moguce ekranski prikaz podijeliti na osam prozora u kojima se instrukcije LIST i PRINT mogu normalno koristiti Definiranje prozora je vezano uz instrukciju WINDOW kojoj se pridružuje 5 parametara Prvi parametar je kanal ili broj toka podataka, e ima znacenje indentifikacija odgovarajuceg prozora koji moze biti odabran PRINT instrukcijom Ostala 4 parametra odreduju granice prozora koristaci uobičajeni koordinatni sistam računala CPC 464 Prva dva od cetiri parametra odreduju širinu prozora (lijevu i desnu) granicu a druga dva parametra odreduju visinu odnosno donju i gornju granicu

Primjer

```
WINDOW  1,15,25,20,30
```

Dakle 1 je broj kanala širinske koordinate su 15 i 25 a visinske su 20 i 30

Točan proator akrena koji se zauzime ovisi o modu koji se upotrebljava odnosno numeriranje koordinate je različito za svaki mod Uz instrukciju WINDOW postoji instrukcija WINOOW SWAP koja se koristi za mijanjanje broja kanala izmedu dve prozora npr

```
WINDOW SWAP 2,3
```

znači da će se podaci kanala 2 prenijeti na kanal 3 i obrnuto Ta instrukcija se moze koristiti za usmjeravanje podataka s , glavnog akrena (kanal 07) preme prozoru, npr

```
WINOOW SWAP 0,3
```

znači da se podeci s ekrana salju na kanal 3 odnosno na prozor što se u programima može iskoristiti na razne načine

JEDNOSTAVNA ANIMACIJA

Amstrad/šnajderom sa moze izvoditi vrlo kompleksna animacija ali svakako treba prvo shvatiti bil te animacije na jednostavnim primjerima

Princip animacije je jednostaven — štampa se neki znak (karakter) na ekranu a potom se isti taj znak štampa na drugom mjestu s time da se prethodni briše Također je vrlo lako animirati objekt sastavljen od više znakova u vodoravnoj liniji

Primjer

```
 15 REM pokretanje
 25 CLS
 35 pokret$=CHR$(32)+CHR$(205)
      +    +CHR$(204)+CHR$(32)
 45 a=17   b=24
 55 WHILE 1
 65 IF NOT INKEY(71) THEN a=a-1
 75 IF NOT INKEY(22) THEN a=a+1
 85 IF a<1 THEN a=1
 95 IF a>34 THEN a=34
105 CALL &BD19
115 LOCATE a,b   PRINT pokret$
125 WEND
```

Pokretanje objekta po akranu je moguce tipkom z ulijevo ia tipkom / udesno Te tipke su određena programskim linijama 65 i 75 Linije 85 i 95 sprječavaju bjezanje objekte koji se pomiče izvan ekrana Linija 105 predstavlja subrutinu operativnog sistema koja odreduje čekanje stroja na spis nove video-slike Konkretno time je izbjegnula dupla slika objekta pri kretanju

Vertikalno pomicanja se moze prikazati sljedecim primjerom

```
  5 REM vertikalno pokrtanje
 15 CLS
 25 RANDOMIZE TIME
 35 a=1   b=1
 45 LET C=INT(RND(30+5))
 55 FOR d=1 TO 23
 65 LOCATE c,d   PRINT CHR$(228)
 75 LOCATE a,b   PRINT CHR$(32)
 85 a=c   b=d
 95 NEXT d
 99 GOTO 45
```

Programska linija 25 postavlja generator slučajnih brojeva Linija 35 postavlje inicijalne vrijednosti dvija varijable koje se koriste za pamcenje posljednje pozicije Linija 45 postavlja horizontalnu poziciju objekta Petljom FOR NEXT kontrolira se kretanje Kontrolna varijabla se koristi kao vertikalna koordinala Linija 65 prikazuje objekt a 75 ga brise s prethodne pozicije

Ovime su prikazana neke najjednostavnije osnove grafike na racunalu CPC 464 Potpuno savladavanje grafičkih mogućnosti zahtijeva mnogo vjezbe ali trud ce se isplatiti jer ovo racunalo predstavlja vrlo dobar kisi za crtanje

Zvonimir Vistrička dipl ing

Programiranje na mašincu | kodovi iz rukava

Spektrum

Postoje mnoge knjige koje se bave Z80 procesorom. Postoji, na žalost, i mnogo onih kojima je strana literatura iz ovog ili onog razloga teško dostupna. Najzad, postoje mnoge male tajne procesora Z80 koje se otkrivaju dugogodišnjim radom, a koje ni u jednoj knjizi nisu opisane. Ovaj članak je, pre svega, namenjen onima koji su, uz teška muka, već usvojili osnove mašinskog programiranja, i koji su zadovoljni onih malih sitnih trikova ima ih bar milion — koji omogućavaju pisanje kompaktnog i efektnog koda.

Kako obrisati akumulator

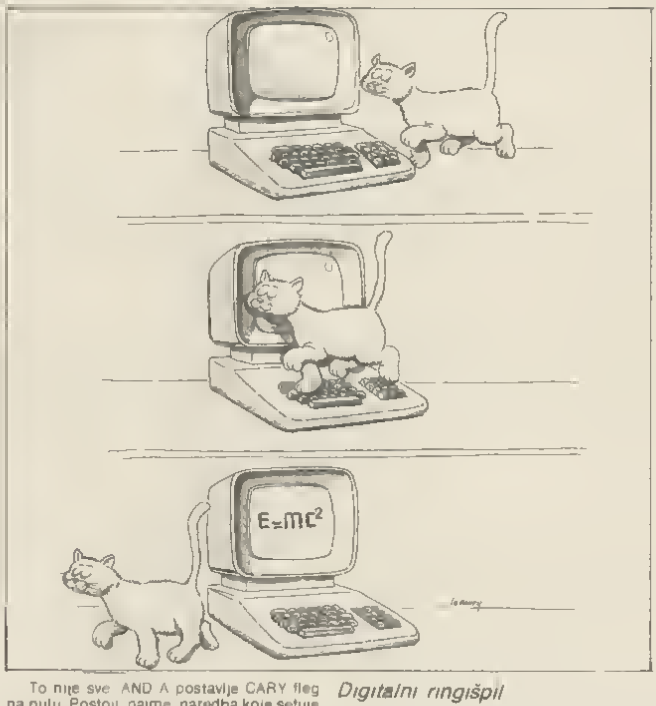
Krenimo od naredba XOR. Radi se o ekskluzivnom OR-u, što će reći da 0 XOR 0 daje nulu, baš kao i 1 HOR 1, dok dve preostale kombinacije (0 XOR 1 i 1 XOR 0) daju jedinicu. Dakle, prosto rečeno dva iste daju nulu a dve različita jedan. Seme naredba XOR s znači izvrši ekskluzivno OR nad akumulatorom i operandom s (neki registar, broj, podatak (HL), (IX+d) ili (IY+d)) a rezultat smesti u akumulator. Pri tome se prvo uzme bit 0 akumulatora i bit 0 operande, izvrši XOR i rezultat stavi u bit 0 akumuletora. Zatim se pretazi na bit 1, 2, 3 i tako redom sve do sedmog. Na kraju se CARY fleg (C) postavi na nulu a ZERO fleg (Z) ne jedinicu ako je novi sadržaj akumuletora jednak nuli, odnosno na nulu ako nije.

Lepo je sada XOR A. Prvo moramo da primetimo da se u ovom slučaju XOR operacija vrši sa dva potpuno ista operande — sa akumulatorom i akumuletorom. Zamislimo za trenutak da je bit 0 akumuletora jednak nuli: 0 XOR 0 daje nulu. Neka je sada bit 0 jednak jedinici: 1 XOR 1 takođe daje nulu. Dakle, ma kakav bio sadržaj akumulatora, XOR A garantovano daje nulu.

Zar nismo mogli da kažemo LD A, 0? Pa mogli smo. Ali LD A, 0 zahteva dve bajte i izvršave se za sedam otkucaja kloka. XOR A zahteva samo jedan bajt i izvršave se za četiri otkucaja. Zašto razbacivati vreme i memoriju? Uzgred rečeno, neki autori više vole da koriste SUB A, sa potpuno istim efektom — SUB A znači od akumulatora oduzmi akumulator i rezultet smesti u akumuletor.

. . . a kako C fleg

Naredbe AND A ima potpuno istu namenu, štedi vreme i memoriju. Potpuno je identična naredbi XOR s tim što se vrši logičke operacija „i" — 1 AND 1 daje jedan. Sve ostale kombinacije daju nulu. Ako seda uzmemo da izvršimo AND operaciju sa dva potpuno iste operanda, rezultat u našem slučaju akumulator ostace potpuno nepromenjen. AND A nema nikakvog dejstva. U stvari ipak ga ima: ZERO fleg će biti setovan ako se u akumulatoru nalazi nula. Umesto da sadržaj akumuletora ispitujemo sa CP 0 možemo da kažemo AND A i uštedimo jedan bajt i tri otkucaje kloka.



To nije sve. AND A postavlje CARY fleg na nulu. Postoji naime naredba koje setuje CARY fleg ali ne i ona koja ga resetuje. To se istina može postići sa SCF CCF, ali XOR A deluje mnogo elegantnije i brže.

Da li je u registru nula

Što se ostalih osmobitnih registara tiče, da bi proverili da li je neki od njih na nuli možemo da upotrebimo sledeću caku: INC s DEC s. Prvo se registar povecе za jedan, e zatim smanji za jedan. Pri tome će, naravno, zero fleg biti setoven, eko se u registru nalazi nula. Prosto, zar ne?

Digitalni ringišpil

A sada ciklusi. Suprotno vrlo rasprostranjenom mišljenju sa DJNZ naredbom se mogu realizovati ciklusi i do 256 e ne samo do 255. Tajna je u tome što se nula može interpretirati i kao nula i kao 256. Ako je potreban ciklus koji ide preko 256 onde DJNZ više ne pali pe se trebe setiti nečeg boljeg. Prvo pede na pamet de se dva ciklusе ostvarene pomocu DJNZ upekuju jedan unutar drugog. Lepo ali može i oveko

```
DEC BC
LD A,B
OR C
JR NZ, LOOP
```

BC registarski par služi kao brojač. Prvo se BC smanji za jedan. Na izuzetno veliku žalost naredba DEC BC neće delovati na ZERO flag pa treba nekako ispitati da li je BC stigao do nula. Prvo sa sadržaj B registra prebacuje u akumulator (LD A,B). Sada treba proveriti da li su i A i C na nuli. Zašto da ih na saberemo sa ADD A C pa ako rezultat nije nula onda treba ponoviti ciklus! To je u redu ali sa na žalost ne može primeniti prosto sabiranje. Šta ako je na primer A = 130 a C = 126? Dobija se 256 a i to je nula!

Primenićemo logičku operaciju OR nad akumulatorom i registrom C (OR C). Pri tome dve nule daju nulu a ostala tri moguće kombinacija jedinicu. Dakle, ako je bilo koji bit akumulatora ili registra C jednak jedinici rezultat sigurno neće biti nula. Provera je gotova ciklus radi. Primetimo da će se stvar vratiti 65536 puta ako je na početku ciklusa u BC atavizmo nulu. Jedina nevolja sa ovim metodom je u tome što uništava sadržaj akumulatora.

Postoji zato jedan drugi metod koji omogućava cikluse do 32767

```
LOOP DEC BC
     BIT 7 B
     JR Z LOOP
```

Broj 32736 je petnaestobitni što će reći da je šesnaesti bit (BIT 7 B) jednak nuli. Pretpostavimo sada da je postepenim smanjivanjem taj broj stigao do nule. Još jednom primenimo DEC BC i iz nula se rađa 65535. Šesnaesti bit više nije jednak nuli i to je znak da je ciklus okončan. Prosto jednostavno, genijalno! Samo primetite sitnicu kriterijum za izlaz iz cikluse više nije nula kao u ranijem primeru već −1 (65535) što znači da će se ringlšpil ovaj ciklus obrnuti jedanput više nego predviđeno. Da bi se to izbeglo pred početak ciklusa u BC registarski par treba staviti broj ciklusa umanjen za jedan.

A zašto na mašinskom jeziku ciklusi skoro uvek idu a brda nadole? Zar nije lakša da idu nagore sa korakom 1, kao na bejziku? Nije lakše. Mnogo je jednostavnije ispitati da li je neki registarski par stigao do nula nego recimo do 3160.

Uslovni restart

Da li ste ikada čuli za naredbu RST Z 38? Za neobaveštene to nije ništa drugo nego JR Z,-1. Stvarno sviđaoi Naredba JR Z −1 se sastoji od dva bajta. Prvi je kod operacija a drugi ono−1. Kada mikroprocesor izvršava tu naredbu, prvo „učita" prvi bajt, a odmah zatim i drugi. Programski brojač se zada nalazi pozicioniran na prvoj sledećoj naredbi. Ako je ZERO fleg setovan onda se od PC oduzima 1, i on pokazuje bas na ono − 1 — što će reći 255. Program se odatle nastavlja s 255 je kôd za RST 38!

Alternativni set

Alternativni set registra je jedna vrlo specifična tvorevina. U dobrim rukama stvara čuda i strahovito ubrzava program. Krenimo redom. Pored AF registarskog para postoji još jedan alternativni AF registar obeležen sa AF'. Mikroprocesor može u JEDNOM trenutku da koristi SAMO JEDAN AF registar. Koji će to biti, određuje jedan FLIP-FLOP. Pomoću naredbe EX AF, AF' može se promeniti stanje tog FLIP-FLOP-a. Jedna EX AF, AF' naredba i mikroprocesor radi sa alternativnim akumulatorom. Tada sa sve naredbe tipa LD A a, AND a, OR s, ADD A, a, itd. odnose na alternativni registar AF'. Još jedna EX AF, AF' naredbe i sve se vraća na normalno. Baš kao što postoji alternativni registar AF' tako postoje i alternativni registri BC', DE' i HL'. Naredba EXX naređuje procesoru da „zaboravi" na BC, DE i HL registre i, uključi BC', DE' i HL'. Naravno još jedna EXX naredba i sve se vraća na staro stanje. Prilikom izvođenja naredbi EX AF, AF' i EXX na gubi se nijedan podatak ni u običnim registrima ni u alternativnim. Nije na odmet spomenuti broj 2758 (heksadekadno) koji mora da se nalazi u HL' registru pri povratku u bajzik. U protivnom, sledi krah slatama (ovo važi samo za spectrum).

Čemu zapravo, služi alternativni set? Zamislimo da se dva programa izvršavaju u isto vreme (to jest čas jedan, čas drugi). Prvi program može da koristi „obična" a drugi alternativna registra. To nije sva. Naredbe EX AF, AF' i EXX se izvršavaju vrlo brzo — za četiri otkucaja kloka — i mogu da posluže kao neka vrsta vrlo specifične zamene za spore naredbe tipa PUSH i POP. Jednostavno, umesto sedam registara opšte namene imamo ih 14 i smanjuje se potreba za pozivanjem spora spoljna memorija. Tamna strana medalja je u tome što „projektovanje" programa koji vrlo intenzivno koristi alternativni set nije nimalo lako. Programer se vrlo brzo spotlja pokušavajući da prati stotine preiaska sa ovog na onaj set. Treba biti suludi programer (čitaj haker) pa istrajati.

Registar za osveženje

Evo još jednog egzotičnog registra — to je R (REFRESH) registar, koji u stvari spada u hardverski deo mikroprocesora Z80. Njegova je namena da pomaže pri „osvežavanju" memorije a sadržaj mu se vrloglavom brzinom i bez prekida, menja od 0 do 127. Postoji naredba koja sadržaj R registra prebacuje u akumulator ali ona nije naročito korisna. Još manje je korisna naredba koja sadržaj akumulatore prebacuje u R registar. Kod „spektruma" ta naredba ne može da izazove krah. Čak ni kada se ponavlja u ciklusu (?!). Jedina kakva—takva softverska korist od ovog registra bi mogla da bude generisanje alučajnih brojeva. Mada je vrlo problematično koliko bi ti brojevi stvarno bili slučajni ako se generišu u nekom ciklusu. Sadržaj R registra se uvećava za jedan posle odprilike svakih šest otkucaja kloka.

Kako testirati H fleg

I nešto ne previše mračno za kraj. Pored toga što postoji CARY flag koji reaguje kada rezultat pretekne iz sedmog bita, tako postoji i HALF CARY flag (H) koji reaguje na prenos između četvrtog i petog bita u akumulatoru. Na žalost na postoji naredba koja će ispitati H flag. Kako to izvesti?

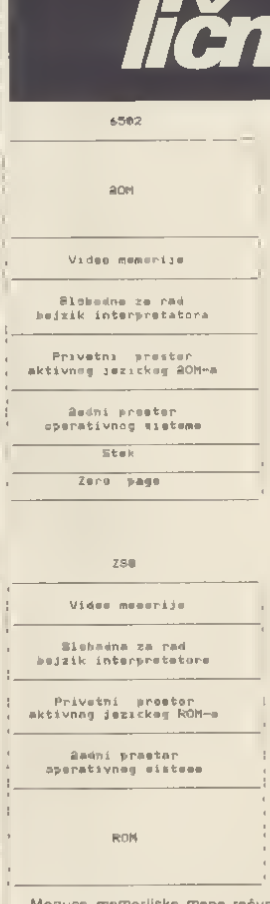
```
PUSH AF
POP BC
BIT 4 C
```

Ko razuma shvatice.

Vladimir Kostic

72) kodovi iz rukava

lične



Moguce memorijske mape računara se Z80 i 6502 su prikazane na slici 1. vidimo da je jedina bitna razlika u tome što Z80 očekuje da ROM zauzima memorijske lokacija počevši od adresa 0 a 6502 za ROM razarviše visoke adrase u našem slučaju počavši od 8000. Na početku RAM-a je radni prostor operativnog sistema i trenutno aktivnog jazičkog ROM-a (deo memorija koji cemo zvati privatni radni prostor bejzika smo dakle, već odvojili jer je bajzik kao što smo videli, cesto samo jedan od raspoloživih ROM-ova). Uveli smo promenljivu LOMEM koja sadrži prvu slobodnu memorijsku adresu iznad radnog prostora operativnog sistema. Od kraja RAM-a smo smestili video memoriju koja u principu može da zauzima manje ili više RAM a zavisno od tranutno izabranog grafičkog

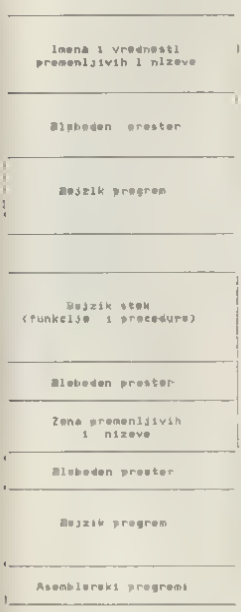
put u središte „ROM"-a/5/

stvari bejzika

Za svoj normalan rad, interpretator bejzika, ili bilo kog drugog programskog jezika, mora da odvoji odradani deo radne memorije u koji će biti smeštana imena i vrednosti promenljivih i nizova, stek za pozivanje potprograma i, naravno, sam program koji se izvršava. Kako organizovati taj prostor da bi se obezbedio minimalan utrošak RAM-a, najjednostavnija konstrukcija samog interpretatora ili, kao poseban žiager, najveća brzina izvršavanja programa.

moda potrebna nam je dakle promenljiva HIMEM koja čuva adresu prvog bajta video memorije. Sav prostor između LOMEM i HIMEM je slobodan za rad bejzika

slike 2:



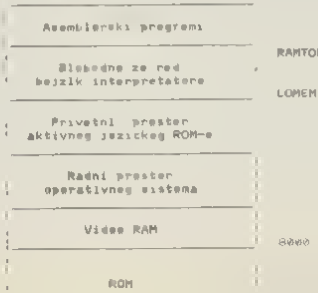
Prostor između LOMEM-a i HIMEM-a možemo da rasporedimo na nekoliko načina od kojih su dva prikazana na slici 2. Prvi od njih je sasvim jednostavan. Počavši od LOMEM-a u memoriju se smešta bejzik program a počavši od HIMEM-a prema nižim adresama vrednosti promenljivih. Uveli smo promenljivu TOP koja sadrži adresu prvog bajta posle kraja bejzika i promenljivu VAREND koja pokazuje kraj zona numeričkih i alfanumeričkih promenljivih i nizova. Samo se po sebi razume da se vrednost TOP-a menja svaki put kada ubacimo izbrišemo ili aditujemo neku programsku liniju dok se vrednost VAREND-a menja kad god definišemo neku novu promenljivu ili dimenzionišemo niz. Ukoliko bi nekim dimenzionisanjem VAREND postalo manje od TOP-a bila bi prijavljena greška tipa No room.

Dobra strana ove organizacija memorija je njena jednostavnost, dok je loših strana daleko više. Prva i osnovna: nismo obezbedili prostor za smeštanje steka koji omogućava povratke iz potprograma. Kad god bejzik interpretator traba da izvrši GOSUB na neko mesto mora da upiše lokaciju instrukcija koju je tranutno izvršavao (i neka druga informacija o kojima ćemo još govoriti) da bi iz potprograma mogao da se vrati. Kako bez potprograma ne možemo da opstanemo, ove informacija ćemo smeštati u deo privatnog radnog prostora bejzika što znači da će korisnik moći da pozova samo nekoliko nivoa potprograma. Ovaj broj je obično šest i dovoljan je za normalan rad ali na omogućava realizaciju rekurzija.

Druga slabost je činjanica da se HIMEM menja pri svakoj promeni grafičkog moda. To znači da će docnije kada u program stavimo instrukiju MODE n biti izgubljena vrednosti svih promenljivih. Što može da buda itekako neprijatno. Mogio bi sa naravno realizovati prepisivanje vrednosti promenljivih ispod novoga HIMEM-a ali je takvo rešenja u slučajevima kada su promenljive ulančane u listu (o toma našto docnija) vrlo neprijatno. Na poslatku ukoliko korisnik našeg računara buda požeiao da piše mašinska rutina naći će se u velikoj navoiji gda da ih stavi? Ako stavija mašinski potprogram iza TOP-a bilo kakvo editovanje bejzika će ga upropastiti. Smeštanje mašinskog potprograma ispod VAREND-a iako moža da dovada do istih posiadica. Ostaje samo da se rutina smesti negda u aredinu memorija izmadu TOP-a i VAREND-a u nadi da ga bajzik program i sadržaji promenljivih neće „pojesti" pri čemu računar neća imati praktično nikakvu mogućnost da datektuje situaciju u kojoj se ovo dašava i prijavi grešku.

Drugo rašenje je daleko bolja ali i komplikovanija za realizaciju. Uvali smo njapre promenljivu PAGE koja označava stvarni počaiak bajzik programa. U počaiku rada računara ja PAGE=LOMEM ali korisnik ima mogucnost da otkuca nešto poput PAGE=PAGE+&1000 i tako razavriše četiri kilobajta za mašinski program od LOMEM do LOMEM+&1000. Kraj bejzik programa kao i do sada označava promenljiva TOP ali sa sada odmah iza bejzika smeštaju promenljive i nizovi. Njihov počatak je promenljiva VARBEG a kraj kao i do sada VAREND. U početku je VARBEG=TOP što znači da će se posle svakog editovanja programa izgubiti vrednosti svih promenljivih i nizova što može ali i na mora da bude problam. ratki su programi koje posle ispravke vradi izvršavati od masta na koma je nastupila greška. Ukoliko je to baš potrabno korisnik ča otkucati našto poput VARBEG=VARBEG+&1000 i tako predvideti četiri kilobajta za produženje bejzik programa.

Od HIMEM-a naniža smo razarvisali prostor za stak pozvanih procedura njegov kraj obeležava promenljiva FRETOP. Prostor izmedu VAREND-a i FRETOP-a je potpuno slobodan i nija ga praporučljivo koristiti za mašinska potprograme obzirom da se prostor za njih razarviše na drugi način. Šta smo dobili ovakvom organizacijom memorije? Bez vačih žrtvi smo pre svega uklonili sve problama koja smo ranije spomanuli. Činjanica je da smo sa odrakim mogućnosti da u nakom od potprograma (kada su naka vrednosti smaštana na stak procedura) manjamo grafički mod tj. utičemo na vrednost HIMEM-a ali je to daleko manji problem nago gubitak vrednosti promenljivih pri pomani moda. Otažali smo na žalost proširanje radnog prostora operativnog sistema ukoliko nam je, na primar potraban dodajni mamorijski prostor za pamćenja definicija karaktara moračamo da pomarimo LOWMEM naviše što će izazvati ne samo gubitak vrednosti promenljivih već i gubitak čitavog bajzik programa. Ovakvi gubici su na žalost naposredna posledica našeg opradaljenja za grafičke modova koji će zauzimali različite kapacitata RAM-a.



Ukotiko ja našoj vibao mamoriji potraban fiksan prostor, možamo da se opredallmo za mapu poput one na slici 3 Vidimo da je video mamorija smaštana zajedno sa privatnim radnim prostorom oparativnog aistama koma, atrogo uzevši i pripada Umesto HIMEM imamo promanljivu RAMTOP koja označava zadnju postojaću mamorijsku adresu Kada nam buda zatrebao prostor za mašinska potprograme III za dafinicija karaktara, otkucaćemo RAMTOP=RAMTOP−&1000 i, kao i ranija imati četiri „skrivana kilobajta Naki računari, kao što ja spectrum imaju naradbu CLEAR kojom se utiča na vrednost RAMTOP-a, dok drugi kao što je TRS 80 zahtavaju da sa valičina ‚skrivane mamorija specificira po uključanju računara

Kako ulančati promenljive

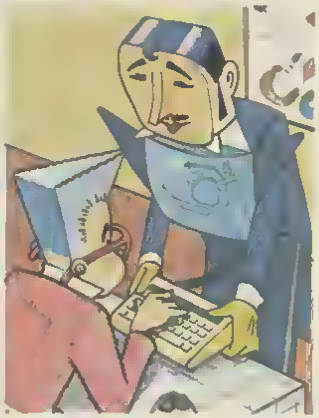
Modul za rad sa promanljivim i nizovima predstavlja relativno složen deo bejzik interpretatora kome ćamo posvatiti dužnu pažnju Rad sa promanljivim traba posmatreti sa dva aspakta možamo da proučavamo način na koji su one reaporedene u mamoriji i algoritme koji mogu da se primene da bi neka od njih bila pronadena ili da razmišljamo o toma kako računar , pamti cela i racionalne brojeve U ovom nastavku našeg putovanja u središte ROM-a ćemo se zabavljati samo prvim aspektom problema dok ćemo preostala razmotriti u sledećim Računarima

Svi znamo šta su bajzik promanljive njihova imana su kod nakada popularnih računara sadržavala samo jadno slovo ili, eventualno slovo i broj dok sa danas za imana promenljivih mogu koristiti i čtava reči Posmatrajmo, najpra najjednoyavniji slučaj u koma postoja samo celobrojna promanljive od kojih svaka zauzima samo dva bajta (vrednosti dakle mogu da se nalaza izmedu −32768 i +32767) i čija ima predstavlja jedno slovo Postoji dakle 26 promanljivih (A, B, C Y, Z) koje sva ukupno zauzimaju 26*2=52 bajta Obzirom da 52 bajta nisu neka alna stavka, razarvisacemo za njih prostor odmah iznad LOWMEM-a dok ća PAGE u startu imati vrednost LOWMEM+52 Modul bajzik interpretatora koji pronalazi promanljivu čija je ima smaštano u čaliju koja sa zova IME izgleda poput slika 4 (izuzali smo dalova modula koji obradu u eventualna graške)

Obzirom da ja ovo prvi modul bajzik interpretatora ili operativnog sistama koji smo u celini napisali možate da pratpostavite da je prilično jednostavan što je i jedina prednost ovakvog statičkog dodaljivanja prostora promenljivima Prva mana koju camo zapaziti ja da smo za numaričke promanljiva odvojili 52 bajta, čak i u stučaju da programer koristi jedino I I J koja bi, same po sabi, zauzele samo četiri bajta Sa druga strana već smo rekli da 52 bajta na predstavljaju naku veliku mamoriju a sada camo priznati da su ta 52 bajta vrlo racionalno iskorišćena nimalo prostore nija utrošno na pamcanja imana promanljivih pošto njih možamo da nadamo koristeći fiksan algoritam Druga mana je što imamo samo 26 promanljivih dok nam može zatrebati daleko više od toga Traca i ubedljivo najozbiljnija mana ovakve organizacija memorije ja postojanja samo jednog tipa promenljivih nama stringova nama nizova i nama matrica! Obzirom da sa čak i najjednostavniji programi na mogu napisati baz korišćanja nizova, možamo da pribagnamo rešenju zastupljenom kod „galaksija počevši od RAMTOP-e u mamoriju sa upisuju alamanti nizs A(1) i to tako da A(0) zauzima poslednja dva bajta mamorija A(1) dva bajta ispred njega I tako dalja U bajzik programu nema potrabe za dimanzionisenjam nizs jer ja on jedinstven i može da zauzme sav prostor od TOP-a do RAMTOP-a dok se adresa njegovih elamenata računaju po formuli **adresa=RAMTOP−2*1+1.**

Statičko dodaljivanje prostora promanljivima ja dakla prihvatljivo samo u slučajevima kada ja prostor u ROM-u jako ograničen i kada je potrebno veoma brz rad računara a prihvatljiv nizak komfor Od njaga na traba, medutim sasvim odustajati mnogi računari (Sharp 1500 BBC B Elactron) odvajaju fiksan mamorijski prostor za 26 celobrojnih promenljivih A%—Z% dok sa ostala promanljiva smešteju u slobodan mamorijski prostor na način o koma ćamo još govoriti Promanljiva A%—Z% (nazivaju sa „razidantna promanljiva) mogu vrlo lepo da sa iskorista za povazivanja bajzik programa sa asamblarskim potprogramima kao i u situacijama kada je potraban brz rad računara

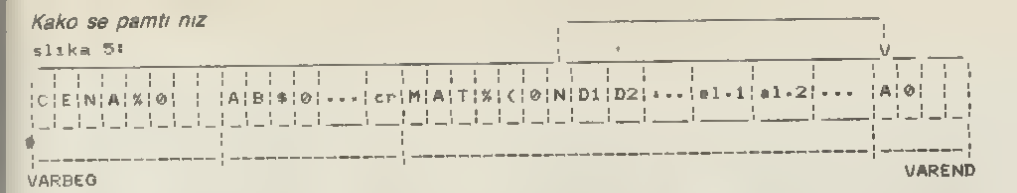
Stedi malo proširenja naših zahteva umesto da dozvolimo samo jednoslovna promanljiva, dozvolićemo imena proizvoljna dužina ali ća vradnosti svih promanljivih i dalje biti cali brojevi koji staju u po dva bajta Realizacija nija naročito problematic na počevši od VARBEG upisujamo ime prva promanljive koja je definisana u programu i to slovo po stovo, koristači standardni ASCII kod Iza poslednjag slova imana stavljamo bajt 0 a sledeća dva bajta razarvišemo za vrednost promanljiva Šta amo na ovaj način izgubiti? Pre svega utrošak memorija za promenljivu A je seda četiri bajta (jedan za slovo A , a drugi za nula bajt iza njaga s dva za vrednost), dok smo kod statičkog dodaljivanja memorija promenljivima trošiti dvostruko manja Osim toga ako sa u bajzik programu pojavi mnogo promanljivih svako pozivanja naka od njh će odnosati zamatno procesorsko vrame Dok za prvi problam ieka praktično nama (mali dobitak postižamo ako na stavljamo nula bajt iza poslednjag slova imana promenljiva, nago satujamo najznačajniji bit tog slova) — jar da bi računar prapoznao promanljivu **ADRESA** ta rač mora da buda nagda upisans na rešavanju problama brzina bi sa dalo poraditi Prs nago što se tima zabavimo, razmisticemo o mogućnostima za smaštanja matrica i nizova kao i skalarnih promenljivih različita dužina (racionalna, celobrojna, atringovi)



```
alika 4:
LD   A,(IME)        ; IME je calija privatnog radnog prostora bejzika u kojoj
                    ; se cuva ima promenljive kojoj treba pristupiti.
SUB  A,ASC "A"      ; Obzirom da je u A ASCII kod slova, traba ga sveati na
                    ; broj izmedju 0 i 25.
ADD  A,A            ; Mnozimo akumulator sa 2.
LD   L,A
LD   H,0            ; Smestamo udaljenost trazene promenljive od VARBEG u HL.
LD   DE,(VARBEG)
ADD  HL,DE          ; U HL dolazi adresa prvog bajta promenljive.
LD   E,(HL)         ; U E ide LSB promenljiva.
INC  HL
LD   D,(HL)         ; U DE je sada vradnost promenljive.
RET                 ; Kraj modula.
```

Kako se pamti niz

slika 5:



Na slici 5 vidimo način na koji se pamte razni tipovi skelernih promenljivih Vidimo, ns primer de sa celobrojne promenljive prepoznaju po znaku za proosnet ne kraju imene, dok stringovi umesto njege eadrže dolsr Ne istoj slici je prikazano de je format u kome se čuve mstrica znetno drugačiji poslednje slovo imena metrice je obavezno leva zagreda, e zatim sledi bajt koji sadrži ofset do početke elemenate metrice pri čemu je **ofset =2*D+1**, gde je D broj dimenzije Iza ofsete sliede dimenzije nizs navedene u DIM neredbi i, izs poslednje od njih prostor rezervisan ze elemente nizs Pogledsjmo keko to izgleda ne primeru ne slici 6 je prikazen bejzik programkoji je dimenzionisao celobrojnu metricu A%(1 2) i dodelio njenim elementime elučejne vrednosti e zatim i heks-dump dele memorije od VARBEG do VAREND uz preteče komentare Izgleda jednostavnije kade se poglede nego kede se o njemu priče zer ne?

```
slika 6:
10 DIM A%(1,2)                          VAREEO=&3880
20 FOR I=0 TO 1                         VAREND=&3013
30     FOR J=0 TO 2
40         A(I,J)=RND(255)
50     NEXT J
60 NEXT I
70 END

3000: 41 25 20                  A%(     ime niza.
3003: 05                        .       ofset do prvog elementa, 3880-3883.
3004: 01 50                     ..      prva dimenzija 1.
3006: 02 00                     ..      druga dimenzija 2.
3008: 05 88                     .       A(0,0)=5
300A: C8 00                     ..      A(0,1)=&C8=200
300C: DC 00                     ..      A(0,2)=&DC=270
300E: 21 80                     !.      A(1,0)=&21=33
3010: 10 C0                     ..      A(1,1)=&10=27
3012: 30 00                     1.      A(1,2)=&9B=59
```

Preostao nam je problem stringova koji su jedine promenljive unapred nepoznete dužine (svaka oslobrojne promenljive uvek zauzima dva ili četiri bajta sveka recionelns promenljiva pet ill šest dok se broj (kilo)bajta potrebnih za pamćenje niza ili metrice može uvek izračunati ne osnovu tipe elemeneta i zadetih dimenzije) String je medutim menje-više prolzvoljne dužine (obično je mekaimelna dužine stringa 256 slova, eli je saavim blisko pameti de nečemo rezervisati toliki memorijski proetor za sveku alfenumeričku promenljivu) pe se izo imene alfenumeričke promenljive obično upisuje njena vrednost, slovo po elovo pri čemu se izo poslednjeg slova stavlja ŠOD (CR tj Cerridge Return) To sesvim lepo funkcioniše eve dok korlanik ne upotrebi nsredbu poput TEKST-$=TEKST$+SLOVO$, kade bejzik interpretator mors de produži jednom definisani string Ukoliko posle definisanje tog stringe nisu definisane nikakve druge promenljive nems problemе ŠOD će jednostavno biti prebrisano deljim tekstom e zatim će ne njegov kraj biti ubeležen novi CR karakter No šta da se redi sko je izvrševen program poput

```
10 I=0
20 A$="Zdravo"
30 INPUT B$
40 A$=A$+B$
50
```

Jssno je de je posle promenljive A$ bilo definiseno B$ i ds neme mogućnosti de sa A$ produži bez pomerenje B$ (već smo rekli da sa pomerenje ovoge tipe obično izbegeveju) Tads će bajzik interpretetor stistnuti zube prebrisati sadržaj promenljive A$ e zatim formirati novu promenljivu sa istim imenom ne vrhu steka I dodeliti joj odgovarejucu vrednost Prostor izmedu promenljivih I i B$ će ostati prazan i nekorišten Ukoliko se ovakve situecije dogedaju reletivno retko, nestabilizaciono ponašanje može de se preželi No, šta eko korisnik napiše program poput sledečeg

```
10 A$=""
20 INPUT B$
30 REM     segment koji pozive potprograma koji definišu promenljive
40 A$=A$+B$
50 GOTO 20
```

Vidimo de se u svakom prolazu kroz petlju ne A$ dodeje po jedno slovo pri čemu sveki put mora de sa formira novo A$ I upropasti po malo memorije Šta de se radi kade je memorije pune? Neki bejzik interpretatori tede stertuju proces koji se popularno nazive „trash colecting ill sakupljenje đubrete Interpretetor vrši potpuno prepekivenje zone promenljivih, teko de nekorišten proetor bude popunjen Moguće bi bilo nerevno vršiti „trash colecting' posle produžavanje bilo kog stringa ell sa pokazalo da je tekve radnja gubljenje vramena u većini primene neča ni doći do prepunjenje memorije Obzirom de je „trash colecting pnično neprijetne radnja koje može de dovede do toga de se isti sagmant u različitim prilikame izvršava za bitno različito vreme mnogi bejzik interpretetori jednostavno ne podržavaju ovu opereciju memorija sa puni i kada bude nepunjene prijevljuje sa greška Imao već bejzik interpretetor tresh colecting ili ga nemao nećete pogrešiti eko na početku programa definišete stringove ne njihove meksimalne dužine Neredba A$=STRING$(80,"") ime amisla čak i ako će odmah iza nje sledili neredba A$="" I

U znaku brzine

Program sa mnogo promenljivih eko je njihovo pronelaženje organizovano ne nečin koji smo opisali neme mnogo šena de bude šampion u brzini Vešt programer će se sneći definišući ne samom početku bejzik programa promenljive koje će se često koristiti eli je tekvo rešenje kao i sva rešenje koje traže od progremera da pozneje konatrukciju interpretetore ze dobrog konstruktore neprihvetljivo Zbog toge ćemo pokušati de ubrzamo pretraživanje zone promenljivih eli je pre toga neophodno razumeti de se nikakav znečajen dobitak u brzini ne može dobiti bespletno biće neophodno ne samo više prostora u ROM-u za progreme koji će operisati sa promenljivime već i više bejtove RAM-e ze njihovo ameštenje Obzirom de cene memorijskih čipove dense nieu prevelike trampa memorije za brzinu se obično pokazuje unosnom

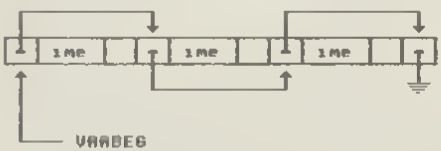
Umsato de prosto nebrojimo imane promenljivih i njihove vrednosti možemo ova elemente de ulenčamo u listu Slog koji se odnosi ne promenljivu koje je u programu najpre definisana počinje od adrese VARBEG Prva dva bajta tog sloga sadrže adresu sloga koji se odnosi ne drugu promenljivu sledi ima prve promenljive i prostor predviden za njenu vrednost Kade bejzik interpratator docnlja bude tražio TEKST$ neće moreti da pretražuje sva prethodne stringove bajt po bajt de bi prepoznao kraj jednog i početak drugoga čim primeti da ima promenljiva ne koju je nešao nije TEKST$ znače gde se nalazi sledeći kandidet! Poslednji definisani element liste sadrži dva nule bajta umesto ukazatelja na sledeći element što čini algetemsku promenljivu VARENDnepotrabnom Čitava ovakva organizacije je šematski prikazana ne slici 7 Posmetrajući je, primećujemo zbog čega je problem premeštenje zone promenljivih na neko drugo mesto (ako je, pošto se bejzik program produžio, potrabno promeniti VARBEG) neprijetan i komplikovan ne samo što bi bajzik interpretator morao de pomera blok memorije, već bi morao da prilagđave zaglevlje svekog eloge Ovo prilegođevenje bi se moglo otekšati kada bi se umsato dvobajtna edrese uveo jednobajtni ofset, odnosno relativno rastojanja do sledeće pro-

→

manipulative ali bi sa tada pojavio problem sa matricama koje zauzimaju više od 256 bajta. Primetimo osim toga koliko je trash collecting dugotrajan i težak za realizaciju.

bolje koristiti takozvane **hash funkcije**. Uzmimo jednostavan slučaj kada god treba da pronađemo ili formiramo naku promenljivu sabracamo ASCII vrednosti slova iz

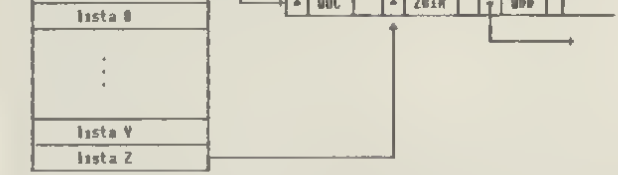
slika 7:



Ulančavanjem elemenata u listu se ostvaruju određeni dobici ali je njeno pretraživanje i dalja zamatan posao dobitak je jedino što na moramo da rešavamo problem stringova promenljiva dužina. Međutim mala varijacija ovog principa može da da izvanredna rezultate umesto jedne formiraćemo 26 raznih lista pri čemu su u svaku od njih ulančana promenljiva koje počinju istim slovom. Nagda u privatnom radnom prostoru bejzika sada moramo da odvojimo 52 bajta za ukazatelja na početak listi promenljivih koja počinju slovima A B C Z. U alogu koji pripada prvoj promenljivoj čije ima počinja sa A nalazi se ukazatelj na sledaću promenljivu koja počinja istim slovom i tako sva do poslednja koja sadrži ukazatelj 0. Ukoliko nija definisana nijedna promenljiva koja počinje slovom K odgovarajuća dva bajta u tabeli iz privatnog radnog prostora bejzika trabe da sadrža nula. Čitava ova koncepcija je shematski prikazana na slici 8.

njenog imena (za ABD$ bi ovaj zbir bio 65+66+68+36=235) a zatim naći ostatak pri deljenju tog zbira sa 26 (u našem slučaju taj je ostatak 1) i na taj način dobiti broj liste u koju je ulančana naša promenljiva. Negda u privatnom radnom prostoru bejzika ćemo naći ukazatelj na ovu listu a zatim pretražujući je i našu promenljivu. Zgodna strana ovoga metoda je u toma što ne moramo da formiramo 26 lista već onoliko koliko želimo da smo želeli 10 lista jednostavno bismo pronašli ostatak pri deljenju 235 sa 10 i tako dobili broj liste. Nezgoda je jedino u toma što ASCII kodovi ne zauzimaju sve vrednosti od 0 do 128 što znači da će naka lista biti opterećanija od drugih. To možamo da izbegnamo uvođenjam naka složanja hash funkcija ali pri tom na traba preterivati izračunavanja vrednosti ova funkcija može da odnasa dobar deo vramena koje je dobijano njanom primanom!

slika 8:



Ukoliko sastavite program koji realizuje ideje iz prethodnog pasusa i isprobate ga, bićete zadivljeni rezultatima toliko da nikada nećete požalati da dalje komplikujete organizaciju promenljivih. Posvetite nam ipak još malo vramena iako objektivno izvanredan, izloženi metod ima mana subjektivne priroda. Pregledajte sve programe koje ste u životu napisali i izbrojte ava imena promenljivih koja počinju slovima Q. Varovatno nećete naći ni jednu. Pokušajte sada sa promenljivima koja počinju slovima A I i J — brojanja će vam brzo dosaditi! Ljudi jednostavno imaju običaj da daju slična imena promenljivima i tako u nakom graničnom slučaju iskorista samo jednu ili dva od raspoloživih 26 lista. Zato ja nešto

Rad sa potprogramima

Rasporedujući memorijski prostor između LOMEM-a i HIMEM-a rekli smo da se na kraju memorije nalazi prostor razervisan za rad sa potprogramima. „Bejzik iz Neolita" je podržavao samo jedan način za pozivanja potprograma — naredbu GOSUB. Obzirom da ova naredba na podržava rekurzija i na omogućava pranošenja argumanata nija potrabno realizovati pozivanja viša od pet-šest nivoa potprograma, što može da se ostvari statičkim dodaljivanjam memorijskog proatora u privatnoj zoni bejzik interpretatora. Modarni bejzici, sa druge strana omogućevaju pranošenja argumanata u potprograma, pa sa interpratator mora više polruditi. Zato ća sa, pri avakom pozivu naka procedura, na bejzik atek (bejzik stak nija isto što i stak mikroprocesora — stak mikroprocesora zauzima deo radnog prostora operativnog siatama, dok se bejzik atak amašta između HIMEM-a i FRETOP-a) smaštati po jedan slog koji sadrže potrabna informacije. U okviru tog sloga se najpre mora smestiti ukazatelj na sledeci slog (nula ako je slog posladnji u listi), zatim zapisati adresa mamorijske ćelije u koju je upisan kraj naredba koja je pozvala procaduru ili funkciju, zatim prapisati čitav stak mikroprocesora i na kraju vrednosti svih stvarnih argumanata i lokalnih promenljivih ta procadura.

Smisao smaštanja adrese kraja naredbe koja je pozvala procaduru na bejzik stek je jasan od log mesta će se po završetku izvršavanja potprograma nastaviti interpretacija glavnog programa. Potraba za pamćenjam lokalnih promenljivih i stvarnih argumanata će biti jasna svima koji pročitaju takst. Funkcije i procedura u ovom broju.

Računara pa ćamo se ovda zadržati na prapisivanju steka mikroprocesora. Ovaj stek dele operativni sistam i bejzik interpretator pri čemu su njegove dimanzije sasvim dovoljne za rad sistema. Bilo bi međutim veoma komplikovano kontrolisati popunjanost steka pra svake PUSH instrukcije (osim naravno ako se za takvu provaru brina sam mikroprocesor) što znači da će prakoračanja kapaciteta steka dovasti do kraha sistema. Bejzik interpratator kao i svaki slozeni mašinski program u valikoj mari koristi stak ostavljajući na njamu brojeve koja će možda i znatno docnija koristiti (u ovu tvrdnju ćete povarovati tak kada u sledaćim Računarima budamo govorili o izračunavanju brojnih izraza). Ukoliko je izmadu stavljanja nakog podatka na stak i njagovog korišćanja pozvana naka procedura koja ja rekurzivno pozvala samu sabe i šazdesetak puta ponovila istu operaciju stak mikroprocesora će se prapuniti sa katastrofalnim posladicama. Da bismo to izbegli, prapisaćamo čitav stak na bejzik stak i zapamtiti vrednost registra SP a zatim inicijalizovati ovaj registar tako da se stak mikroprocesora puni od početka. Kada rad potprograma buda završen registar SP će dobiti staru vrednost stak će biti restauriran i rad interpratatora će se normalno nastaviti. Jedino ograničanja ove koncepcija je činjenica da upotraba steka mikroprocesora u proceduri mora da bude „čista" na može se u toku interpretacije procedura staviti našto na stak da bi to bilo pročitano kada se buda dalje izvršavao glavni program.

Prapisivanja vrednosti stvarnih argumanata na bejzik stak možamo da izostavimo a da tima nešto i dobijemo. Ukoliko vrednosti svih stvarnih argumanata pranesamo na stak i sa njaga ih skinemo po završetku rada procedura, bilo kakve izmana koju je na nakom od argumanata izvršio potprogram će biti poništana što će onamogućiti vraćanja razultata iz potprograma. Zašto bi sa našto ovakvo izbagavalo? Zamislite da autor bejzik programa napiša nešto poput:

```
100 PROCproblam(1 2 3)
110
200 DEFPROCproblam (A B C)
210 A=5 B=7 C=9
220 ENDPROC
```

Bejzik interpratator će pokušati da konstanti 1 dodali vrednost 5, što će izazvati vrlo čudno ponašanja računara o budućnosti. Naki programski jazici (npr. paskal) rešavaju ovaj problam tako što zahtavaju od korisnika da u listi formalnih argumanata specificira koji će od njih biti menjani a koji neća (ukoliko sa docnija toga na buda držao sam je kriv za ono što će se dogoditi), dok drugi (fortran 77) podrazumevaju da

potprogram sme da menja sve argumente, a funkcija ne smi ni jedan. Ukoliko sastavljate bejzik interpretator možete po želji da se opredelite za bilo koju od ovih koncepcija a da vaše rešenje ima jednako mnogo dobrih i loših strana.

Problemi u vezi sa procedurama i funkcijama predstavlja i njihovo pronalaženje kada napišete GOSUB 1000 bejzik interpretator će pretražiti prostor između PAGE i TOP i pronaći linijski broj 1000 (ako on postoji), posle čega će znati gde počinja potprogram. Pri pozivu procedure se navodi njeno ime što je daleko više u skladu sa normama strukturiranog programiranja (kada docnije budete gledali program. značete da PROCx0y crta koordinatni sistem dok vam GOSUB 2788 neće ništa značiti) ali će zato pronalaženje procedure biti zamašan posao. Linije programa su kao što ćemo videti ulančane u listu pa je pronalaženje neke od njih manje-više brz posao dok se traženje nekog imena može realizovati jedino mukotrpnim pretraživanjem zone programa bajt po bajt. Rešenje je mali trik pri prvom pozivu potprograma moračamo da pretrazujemo memoriju dok ga ne nademo. Tada cemo u zonu numeričkih promenljivih (iznad VARTOP-a koji cemo zatim modifikovati) upisati slog koji sadrži ime pronađena procedure i njenu adresu u memoriji. Sve ovakve slogove povezujemo u jednu listu i izdvajamo dva bajta u privatnom radnom prostoru bejzik interpretatora za pokazivač na njen početak. Kada docnije bude pozvana neka procedura proveriće mo da li se njeno ima već nalazi u listi i ukoliko ga nademo eliminisati potrebu za pretraživanjem programa. Dalje dobitke u brzini biamo dobili ako bismo formirali 26 lista po jednu za imena procedura koje počinju istim slovom, ali je ovakvo nešto obično sasvim nepotrebno.

Pakovanje programa

Bejzik program je u stvari običan tekst koga bismo mogli da formiramo uz pomoc nekog standardnog tekst editora (tako se to i radi na velikim sistemima). Bejzik interpretator koji sastavljate mora, medutim da sedrži i neki editor koji će omogućiti nesmetano i komforno (ovu poslednju karakterstiku zaboravite ako planirate da tražite posao u firmi Sinclair Research) kucanja i ispravljanja programa. Pošto je ovaj editor integralni deo bejzik interpretatora i neće biti korišcen ni za šta drugo bejzik interpretator može da na njega prebaci deo svojih poslova pa će sa u memoriji nalaziti tokenizovan bejzik program.

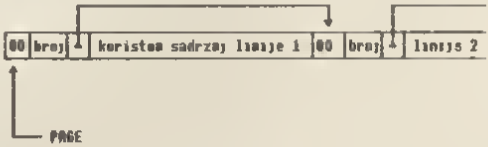
Najjednostavniji editor bi ubacivao linije u memoriju baš onako kako ih korisnik kuca. linija 1000 GOTO 700 bi bila upisana kao na slici 9. tako bi nešto ovakvo bilo sasvim jednostavno za realizaciju. pri izvršavanju programa bi svaki put trebalo pretvarati string 1 0 0 0 0 u linijski broj 1000 i string 7 0 0 u broj 700. Osim toga, traženje linije 700 bi se obavljalo tako što bi mikroprocesor lagano pretraživao program. pronalazio karaktere &0D koji označavaju krajeve linija i prepoznavao linijske brojeve iza njih. Ovakvo je rešenje prihvatljivo samo za računare sa veoma malim ROM-om. malim usložnjavanjima interpretatora mogu da se postignu ogromni dobici u brzini.

Osim naredbi često se tokenizuju i specijalni znaci. Time se. dalje. ubrzava izvršavanje programa. tokeni su obavezno grupisani tako da naredbe koje obrađuje jedan modul operativnog sistema imaju slične tokene. pa ih je moguće vrlo brzo prepoznati. Na ovaj način smo dobili na brzini izvršavanja programa i uštedeli na memoriji koja je potrebna za njegovo smeštanje uz minimalne žrtve. morali smo da usložnimo operativni sistem dodajući module za tokenizaciju i detokenizaciju naredbi.

Osim tokenizacije transformišu se i sve numeričke konstante za bejzik interpretator je konvertovanje stringa 1 234 E15 u broj u pokretnom zarezu zamašan posao koji nije racionalno obavljati pri svakom izvršavanju naredbe. Zato će većina računara iza alfanumeričke predstave broja razervisati četiri ili pet bajtova u koje će biti upisana njegova binarna predstava. pri izvršavanju programa će string jednostavno biti ignorisan. Ako ubuduce negde pročitate da je na spektrumu bolje napisati LET A=VAL "1234" nego LET A=1234 jer prva varijanta zauzima manja memorija imaćete priliku da se osmehnete. Jasno je da autor takve pojave ne zna da se primenom izložena, veština odrekao onoga što su konstruktori spectruma uradili za njega trampe nekoliko bajtova memorije koje spectrum i tako ima dosta za dobitak u brzini reda, koja ovom računaru nije jaća strana.

Ukoliko ste pročitali i razumeli ova četiri nastavka našeg Putovanja u središte ROM-a i možete da kreativno primenite stečeno znanje, nećete imati mnogo problema se razumevanjem strukture i rasporeda modula operativnog sistema i bejzik interpretatora bilo kog kućnog računara. Ukoliko požalite da sami pišete nake od tih modula, potrebno vam je više detalja kojima posvećujemo naše sledeća nastavka.

slika 10:



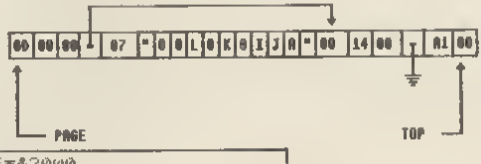
Na slici 10 vidimo da se na početku svake linija nalazi njen broj ali predstavljen u obliku dvobajtnog celog broja, a ne u obliku stringa. Sledi bajt koji sadrži dužinu linije što znači da bejzik program možemo da zamislimo kao ulančanu listu linija. Sledi koristan sedržaj linije i na kraju &0D što označava njen kraj. Na prvi pogled bi sa reklo da je ovaj bajt nepotreban pošto je dužina linije navedena na njenom početku. Postojanjem ovoga bajta je medutim, sigurnosni ventil koji ce popustiti čim program bude učinjen neupotrebljivim naopraznom upotrebom naredba POKE bejzik interpretator će u odredanim prilikama proveravati da li se na kraju svaka linija nalazi CR i prijavi grešku ako to nije slučaj.

Što se tiče dela linije koji smo na prethodnoj slici nazvali koristan sedržaj, on se neće sastojati od alfanumeričkog teksta naredbi primenice se tokenizacija. Svaka prepoznata bejzik naredbe ce biti zamenjena jednim jedinim bajtom tokanom te naredbe. Tako ce na primer naredbi PRINT biti dodaljen token &87 a naredbi END token &A1 pa ce bejzik program

```
10 PRINT  Galeksija
20 END
```

biti smešten u memoriju kao na slici 11

slika 11:



slika 9:

```
                                PAGE=&2000
                                TOP= &200B

2000: 31 30 30        1000      broj linije.
2003: 20                        blanko pre naredbe
2004: 47 CF D4 CF     GOTO      tekst naredbe.
2007: 20                        blanko izmedju GOTO i 700.
2008: 37 30 30        700       adresni deo naredbe.
200B: 0D                        kraj naredbe.
```

Daleko vam je, naravno, potrebnije iskustvo koja jedino možete steći ako zagrejete stolicu računar i monitor.

Dejan Ristanović

rom od sedam milja (2)

Računari u domaćoj radinosti

U listingu izmena u osnovnom ROM-u načinjene su dve štamparske greške. Obe su u delu za objektni kod dok je deo sa izvornim kodom (mnemonicima) sasvim u redu

```
štampano
    1235  21 46 5E    LD    HL,&5E53
traba da glaai
    1235  21 53 5E    LD    HL &5E53
štampano
    04B2  32 CO 5C    LD    HL,&5CCO
traba da glaai
    04B2  21 CO 5C    LD    HL &5CC0
```

Projekat megaroma za spektrum uneo je dosta uzbuđenja u već pomalo učmao život YU spektrumovaca koji su počeli ozbiljno da strahuju da ih vreme polako gazi zajedno sa njihovom mašinom. Bilo je medutim i protekla lipa zaslo nije bilo ni reci o NMI i ostalim bagovima u ROM-u ili zaslo je uvaden link za bejzik a nije se nimalo mislilo na mašinske programere.

Članak uistinu nije bio posvecen mogucim prakrajanjima ,spektrumovog ROM a za sledeci broj pripremamo pregled ideja i praktičnih rešenja mogucih zakrpa — vec samo jednom praktičnom projektu od koga upravo programeri u mašincu imaju najviše koristi. Primadbe ovoga lipa očigledno ukazuju na to da su ovakve izmene najčešce rezultat snažnog poriva da se računar prilagodi sopstvenim potrebama a ona ratko kada mogu biti od opšteg interesa. Žalo uostalom pomocni ROM ovi i postoja — u njih može staviti svako što god hoce.

Ambicioznija vlasnika spektruma najviše je zbunila nekompatibilnost resenja sa mikrodrajvom. Da li je bilo baš toliko komplikovano da sa ona očuva? Nimalo. Za ona koja na zanima mikrodrajv to predstavlja najalagantnije resenje a oni koji ga imaju mogu sa dve-tri intervencija — pod uslovom da smisle zaista bezbedno mesto za ROM OS (operativni sistem za rad sa alternativnim romovima) — da povrale kompatibilnost veoma jednostavno.

- umesto adresna linija A4 na nožicu 1B integrisanog kola 74LS139 traba dovesti liniju A7. adresa za uključivanja eproma ce sada bit 127
- instrukcije lipa OUT (239). A treba prepravit u OUT (127), A
- program treba disasemblirati i promaniti mu pseudonaredbom ORG tzv. origin (odreduje mesto na kome se nalazi i izvršava program u memoriji)

Sledeca grupa pitanja odnosila sa na drugi link za naradbe. Odakle se on poziva? Iz programa koji trajno zauzima prvi link. Nimalo medutim na smela da se i on poziva iz ROM a i to odmah nakon pozivanja prvog linka. Tada bi izmena 4 mogla da glasi

```
0008 CD B6 5C CALL &5CB6
000B CD B9 5C CALL &5CB9
000E 18 43 JR &53
```

Zbog nedostatka prostora na samom prelomu lista morali smo da izostavimo labelu sa strukturom zaglavlja — lične karte koja prethodi svakom programu u epromu. Formal zaglavlja nije isti za sva lipove programa. Ako sa u apromu nalazi samo bajzik ili samo program na mašinskom jeziku onda zaglavlje ima 22 bajta

apsolutna adresa	relativna adresa	funkcija
0000	IX – 01	Broj aproma (0-FF)
0001	IX + 00	Tip programa 00 bejzik 01 bejzik + mašinac 03 mašinac
0002	IX + 01	ima programa (najviša 7 alova), nalakorišcana bajtove popuniti sa FF
0009	IX + 08	Oznaka kraja imana (FF)
000A	IX + 09	Dvobajtna adresa u EPROM-u sa koja traba preslikati program
000C	IX + 0B	Dvobajtna dužina programa
000E	IX + 00	Adresa na koju traba preslikati program za bejzik programa broj linija od koja se program startuja
0010	IX + 0F	Dužina programa sa promanljivim (bejzik) nova adresa mašinskog staka (mašinac) FFFF ako za mašinac na trabe nova adresa staka
0012	IX + 11	Adresa sledećag zaglavlja u apromu ill FFFF
0014	IX + 13	00 u EPROM u nama više programa FF u EPROM-u ima još programa
0015	IX + 14	00 program se na nastavlja u drugom apromu. NN program sa nastavlja u apromu NN
0016	IX + 15	Prvi bajt mašinskog programa III bejzika Za kombinacija bejzik/mašinac i za programa koji sa nastavljaju u drugom apromu izvorna adrase mašinskog bloka ill nastavka
0018	IX + 17	Dužina mašinskog bloka ili nastavka
001A	IX + 19	Odradišna adresa mašinskog bloka Ili nastavka
001C	IX + 1B	Adresa novog mašinskog staka ili FFFF
001E	IX + 1D	Prvi bajt programa

Ako se medutim u njemu nalazi kombinacija i bejzika i mašinskog jezika ili ako se program nastavlja i u drugom apromu onda ima osam bajtova više dakle 30.

Prvo zaglavlje uvek počinja na adreai 0000 a sledace ako u apromu ima još programa na adresi – dužina prvog programa + dužina njegovog zaglavlja. Adresa zaglavlja stalno se čuva u registru IX a do svih potrabnih podalaka se dolazi preko indaksa. Prvi bajt u zaglavlju (IX – 1) u ovoj varziji programa ROM OS nije iskoriscan — njime je samo pripremljen teren za savršenju verziju koja ce izmedu ostalog, imati i naradbu *CAT(alog). Na pojedinim adresama u zaglavlju nalazi se na istom mestu nekoliko raznorodnih podalaka. Program ROM OS se medutim nikada neca zbuniti — potrebna grananja nalaju na osnovu podatka o tipu programa u epromu na relativnoj adresi IX.

Konstruktorima koji su se prošlog lata u ovo vreme upustili u samogradnju programatora aproma život nija bio nimalo lak — cena inlagrisanih kola su divljala na svajskom tržistu skačuci i do 700 odsto! Ovoga lata duvaju znatno povoljniji vetrovi — cene epromima od 8 i 16 kilobajta su toliko pale da su se izjednačile sa cenom najobicnije igre na kaseti. Tako eprom lipa 27128 sada košta ispod 30 maraka. Ova okolnost verujamo čini naš projekat još privlačnijim.

J. Regasek

Računari u domaćoj radinosti

brisač eproma

Brisanje eprome može da bude izuzetno neporno i složeno, nerocito eko nije ne raspolaganju odgovarajući pribor. Keo sredstvo za brisanje služi UV (ultravioletne) lampa, koje mora de ispuni nekoliko elementernih uslove — treba de emituje ultraljubicaetu svetloet odredene talasne duzine (254 m) i de ima energiju zracenja od 12mW/cm². Od domaćeg meterijale i u domaćoj radinosti može se, za oamo 2000 dinere, napraviti brisač eprome koji u potpunosti zadovoljave ove standarde.

Najupornij konstruktori koji su u potrazi za lampom za brisanje eproma stigli čak do Namačka i Engleska i vratili sa odande praznih ruku bice bez sumnje iznenadeni našim otkrićem da se ,toliko željena tam pa može naci ne policama svaka vace prodavnica alaktrotehnička robe Nija medutim čudno da je toliko dugo ostajala neprimacana — dobro je kao bisar u školjci skrivena u naprozirnom sijaličnom balonu! Sijalica se može neći pod komarcijalnim imanom Živine sijalica visokog priliska VTFE tipa

Na slici 1 se vidi izgled VTFE sijalice ispod balona za nas je taj deo izuzetno značajen jar se u njamu nalazi „mala sijalica — žižek Slika 3 prikazuje da samo zizak emituje ultraviolatnu svatlost koja je neophodna za brisanje De bi se do njaga došlo pažljivo da otklonite semo spoljni balon — razbijte ga ili odrežite brusilicom — e da pri tom svi unutrešnji dalovi ostanu u onom stanju u kakvom su i bili

Sijalice se ne sma povezati direktno ne 220 V jer cets tada morati u rednju po — novu Keo što se vidi na shemi 2 vezivanja predspojna sprave (prigušnice) i VTFE je serijsko (redno) Kompanzacioni kondenzator nacrtan isprakidano koji je vezan paralelno u odnosu na mrežni napon nije obavezan U obzir dolaze tri tipe sijalica za 80 W 125 W i 250 W Mi vam preporučujemo de odabarete sijalicu od 80 W zbog niže cana i naročito manjag toplotnog efekta

Uz sveki tip sijalica dolazi i odgovarajuca prigušnice dimenzionisana prema odgovarajucoj snazi upotrabljene VTFE-sijalice Prigušnica sa obelazava sa PPO (nazivna snaga) sijalice — za sijalicu od 80 W odgovarajuca prigusnica je PPO80 Ako imate sijalicu VTFE od 80 W a ne možete da pronadete odgovarajuću prigušnicu sevsim dobro ca posluziti i ktesična prigušnica za _neonku od 65 W Za 2000 dinere i pola časa rada konstruktori dobijaju uredaj koji se po atikasnosti može meriti sa bilo kojim inostranim a cena mu je pri tome, nekoliko puta manje



1

Um- mrežni napon 220 V
C kompenzacioni kondenzator
Pr prigušnica

Opasne krivine

• Uradaj krija dva potancijatna opasnosti od kojih sa moramo dobro čuveti Sva delove koji su pod mrežnim naponom potrebno je dobro izolovati a lampu postaviti u potpuno zalvoranu kuliju jer je zracanja opasno za vid (ovaj tip svallosnog zracanja je identičan sa zračenjem koje se stvara prilkom alektrolučnog zavarivanja) U okolini svetilike se oseca i (inače potpuno bezopasna) pojačana jonizacija vazduha

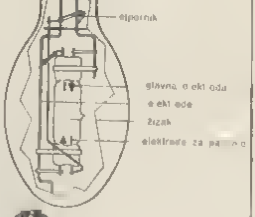
• Razmak izmadu dva paljanja traba da bude oko 5 minuta a lampa postiže maksimalnu jačinu svetlosti za 4—6 minuta

• Eksperimentiše sa rastojanjam izmadu lampe i eproma tako da vrama brisanja buda izmedu 15 i 20 minuta Autor je izbrisao nakoliko tipova eproma za računar galakcija na rastojanju od 10 cm sa lampom od 80 W (vidi tablicu) za manja od 10 minuta (Najbrže su izbrisani Hitači epromi — 7 min) Optimalan položaj za postavljanja eproma je paralelan u odnosu na osu lampa

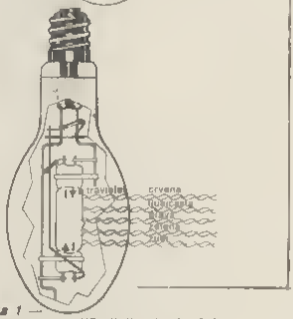
Ukoliko potrebna alamanta ne pronadete u maloprodaji možda ih imaju njihovi proizvodači

TEP Tvornica alaktrolabnićkih proizvoda-Zagreb
41080 Zagrab II Oranički odvojak 18
TESLA fabrika sijalica
26000 Pančevo P Marganovica 14 A

2



3



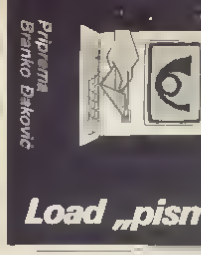
Tabela

rastojanja od podloga	25	50	75	mm
provodni sundar	62 5	32 4	zan	C
aluminijum	48,6	40	zan	C

Zavisnost grejanja kućišta eproma od rastojanje i vrste podloge kada je izložen zračenju 15 minuta merenje je izvršena pri temperaturi ambijanta od 23,2 °C instrumentom „Tektronix" DM 501A

Slika 1 —
Shema spoja VTF sijalice i prigušnice direktna priključivanja na 220 V nija mo ića
Slika 2 —
antaomija VTF sijalice
Slika 3 —
Spaktar zračenja VTF sijalice pošto sijalični balon apsorbuje ultraljubičastu svetlost treba ga pažljive ukleniti

Dragoljub Jokic



Priprema Branko Ekinović

Load „pismo"

Za mase a ne za klase

Javljam Vam se povodom komentara i diskusija brojnih čitalaca kao i intervjua profesora Joza Dujmovica

Činjenica je da mi stojimo ne pragu informacijskog društva i da u ovom postindustrijskom periodu u kome se sada nalazimo moramo podrzati sve napore za širenje računarske pismenosti. Međutim čini mi se da je u sadašnjem tranutku računarstvo u našoj zemlji pomalo popримilo neke odlika pomodarstva. No to je i razumljivo kada nauka nija mogla ili nija imala uslove da preuzma kormilo u širenju računarske pismenosti. Logično je onda da je programerska mladost rekla svoja

Prema mišljenju nekih francuskih sociologa put čovečanstva u informacijsko društvo bice veoma složen. Dok visoko razvijane zemlje Zapada spremno koračaju u poslindustrijskom periodu dotle zemlje istočna Evropa i Jugoslavija još nisu završile ni potpunu industrijalizaciju. Sve ovo je sigurno jedan od bitnih uzroka sadašnjeg stanja razvoja računara kod nas. Mislim da u ovom svetlu treba tumačiti i tekstove drugarice Jelene Rupnik

Što se programskih jezika tiče oni ce sa stalno razvijati i trpeti izmane. Sada je u modi bejzik sutra ce to biti paskal lisp prolog i mikro prolog i drugi jezici veštačke inteligencije a ono što je univerzalno to su algoritmi i algoritamske zakonitosti. Žalo mlade ljude treba učiti algoritmima prama kojima sa pišu odgovarajući programi na raličitim programskim jezicima u zavisnosti od vrste računara kojim sa raspolaze

Pojava mikroprocesora doveta je do računerske revolucije koju sada imamo. Došlo je do automatizacije i robolizacija proizvodnje uvođenjam mikroprocesora i mikroračunera i samo takvom proizvodnjom mi možemo podici produktivnost rada. I dok mikroračunarska tehnologija u Evropi dostiže svoju punu zrelost izlišni su svi razgovori o lomu da li je kod nas suvišna popularizacija računara a posebno personalnih mikroračunara

Prema toma mišljenja sam da je svaka aktivnost koja doprinosi širenju računarske pismenosti u sadašnjoj situaciji za pohvalu

S druge strane smatram da širenje računarske pismenosti treba voditi naučno i organizovano i da bi Univerzitet kao i Republika morali u vecoj meri da utiču na dalje širenje takova kibernetske naučne misli koja obuhvaniuje viša naučnih disciplina teoriju sistema teoriju upravljanja, teoriju informacija teoriju algoritama i teoriju igara kao i njihovu primenu u daljem razvoju društva. Tako ca sve aktivnosti računara kao osnovnih kibernetskih sredstava naci svoje mesto neca sa nakim stvarima davati veci značaj nego što to zaslužuju

Sa ovako koordiniranim aktivnostima nebila zašto nemati poverenja u domaću softver?

Naprotiv treba smalo i organizovano zakoračiti u razvoj informacionih dobara i informacionih tehnologija koje su u Evropi već dobile status novih tehnologija. Takode smatram da treba podsticati razvoj komarcijalno raspoloživih aplikacionih programskih paketa kao i ekspertnih sistema ih oblesti obrazovanja nauke i proizvodnja

Dr ing Jelenka Savkovic Sretenovic docent Tehnološko-metalurškog fakultala u Beogradu

„Amstrad" protiv „amstrada"

Da li ste ikada uspeli da napišete program na „amstradu"? Ja nisam

Dobra osobina mnogih računara kod nas došla je do izražaja tek kada zodg prapunih magacina dobro pojeftine. Tako se na primar za „atari 800XL" nija ni znalo dok mu cana nije drastično snižena. Kralj je jeftin živeo kralj! Nedavno su a prodaji pojavio računar koji za svoju cenu nudi dosta. Vec posle nakoliko meseci u našoj stampi su počeli da mu pišu hvalospeva dižući ga u nebesa. A zna se ono staro ko visoko lati. Pogadate reč je o amstradu

Posle svih ith aplauza koje je pobro na YU sceni kralj amstrad se našao i ne mom stolu. A onda

Na našem softverskom tržištu još uvek ima veoma malo programa za amstrad (a zar ih nagde ima više) pa bi to trabalo da ohrabri (prisili?) programare da i sami na pisu po neki program. Tako sam i ja krenuo da napišem program za svoj amstrad. I to program za crtanje. I računar je počeo da prvi javljuje sintaksna greška. Za divno čudo greška je bila vec u prvom redu 10 PRINT PROGRAM ZA CRTANJE. Lupao sam glavu oko ove linija pola dana i posto nisam uspeo da je ispravim rašio sam da ovaj red izbacim (i lako nije mnogo važan). Međutim računar je prijavio grasku i u sledecoj linji 20 GOTO1000. Katastrofa se ponovila! Rešio sam da se okanem pisanja sopstvenih programa i krenuo da prapisujem programa iz priručnika i gle čuda — oni su radili (!!!) Najzad sam shvatio da sve argumente treba odvajati od naradbi (!?) inače računar odbija da radi. Pomalo zbunjen ugasio sam računar i ostavio sve za sutra. Kakav dan!

Sutradan sam pun eleana ponovo uključio svoj amstrad i učitao program sa demo kasete. Uskoro je slika počela da beži sa ekrana. Ne može to tako uzviknuo sam. Prvo sam osuo drvlja i kamenje na elektrodistribuciju „amstrad" je fina masina smeta mu nastabilan napon! Ugasio sam računar (drugi put) i ostavio sve za bolje dane — do kojih nikada nije ni došlo (posta pola sata rada slika mi je uvek bažala sa ekrana). Bio sam ne tvici da dam oglas. Manjam amstrad za komodor 64 ili spektrum 48K bez doplate

Kasnija sa pokazalo da su to još i najmanja mane amstrada. Velika greška konstruktora je što kod amstrada slatno posloji ekran visoke razolucija za koji se troši 16K RAM memorija u svakom od tri moda. Zahvaljujuci tome od 42K slobodne mamorije za bejzik ostaje svega 26K. To je manje nago čak i kod onog malog crnog računara sa šarenim strafama sa strane!

Kada sam isprobao muzičke sposobnosti toliko hvaljenog čipa AY 3-8910 prilično sam sa razočarao (čip je dobar ati podrška). Od iri glasa koja amstrad ima vi možete da podesite samo jedan a sva iri zvuča isto tako da ne može le napraviti da vam svrauju iri različita instrumanta u isto vrema. Vrlo lose ipak najgora strana amstradovog zvuka je mali zvučnik sme šten negde kod kasetofona. Ton je vrlo los i slab a to opat neodolivo podseča na jedan mali računar

Kad vec pomanuh Ghostbusters za amstrada su se pojavili i prvi programi na domacem (pirat skom) tržištu. Možete ih nabaviti prako Ašlrad Usar Club-a (adresu nadite u malim oglasima) i to po ceni isto kao i u Engleskoj!

Pošto samo pomenuli programe nih treba i učitati u računar. Kod amstrada postoje dve brzine učitavanja 1000 i 2000 boda. Vrlo brzo rekli biste vi. Ne ne kažem ja. Kada učitavate programa brzinom od 2000 boda onda učitavanje istom brzinom kao kod spektruma koji učitava 1200 boda! Rešenja je jednostavno. Amstrad smatra i učitava u blokovima od po 256 bajtova. A pra svakog bloka imate heder u koma je zapisano sva i svašta tako da za program od jednog kilobajta imate četiri (!!!) hedera. Za svaki sledeci kilobajt još četiri pa tako u beskraj. Znači kasetofon olpada kao spoljna periferija. Oslao nam je disk i opet greška. Diskata od 3 inča izgubila je bitku sa onom od 3.5. Čak se i u Englaskoj sve teža nabavljaju diskate od 3 inča. A što je ratko to je i skupo. Od ostalih perifarija amstrad podržava još i printar. U priručniku piše (a i u reklamama) da to radi preko Centronix interfajsa. Mani to mnogo više liči na običan EDGE konektor što znači da je priključivanja standardnih printara potreban poseban kabl

Za džojstik postoji samo jedan port — ako hocete da odigrate nešto protiv komšinice moracete da kupite amstrad police koja su dupio skupije od veoma sličnih QUICKSHOOT I

Mnogi kupuju amstrad zbog loga što je najjeftinija CP M masina (C 128 ?) Prvo daju pare za računar i disk a onda otkriju da ne mogu da nabave ni jednu jedinu CP/M disketu od 3 inča. A ako neku ko zna kako i presnimo može se desiti da program ne radi jer od 64K koliko računar nominalno ima za CP/M ostaje jedva polea to za većinu programa nija dosta. Takode 80 slova u redu sve ne vide bas najbolje ni na amstradovom crno-belom monitoru dok je u boji još gora. A šta tek da kažu oni optimisti koji su kupili amatrad sa UHF modulatorom

Za ona koji žele CP/M mašinu na raspolaganju im je komodor 128 ili za rodoljube iskrin partner. Ili mogu da kupe neki računer koji ima CP M kao proširenja. Što se tiča grafika i zvuka mogu sa nebaviti i računeri sa sličnim ili čak i boljim karakteristikama koji su uz to mnogo potpunji na drugoj strani. U svakom slučaju amstrad je dobar ali suviše izvikan računar koji tek traba da sa odkaže — ima i puno boljih od njega i to po sličnoj ceni. Što sa tiče bejzika amstradov je vrlo dro ali i vrlo neprecizan (radi sa samo iri decimale) ali to nija toliko važno ima i važnijih stvari za jedan računar

Nadam sa da sa amstradovci neće ljutiti zbog ovog članka. Vreme je da sa ukaže i na neka amstradove mane koji ima dosta a niko ih ni na spominja iz ko zna kojih razloga! Vrama ja da Amstradovci svhala da nemaju najodlji kućni računar na svetu — ima i puno odlih

Dragan Grbic

Svaka ima svoj ugao iz koga posmatra kvalitete i mane nakog računara. Dužni ama međutim, da akramamo pažnlu na takozvana nedostatka činjenica

1 Amstrad ima 64K RAMa a za bejzik ostaja slobodne oko 43K što saznajama kucajući naredbu PRINT FRE( ) (a ne 26 K)

2 Tačno je da zvuk iz zvučnika koji je ugrađen u računar nije naročit, ali zato postoji mogucnost priključivanja na stereo pojačalo inače zvuk sa iz komodora na maže uopšta čuti (iz televizora može).

3 Amatrad učitava programe u blokavima od 2048 bajta jer posa duja bafer veličina 2K pa iaka tK programa zauzima atpriliks 1/2 bloka (a na 4)

4 Što sa tiča tačnesti prikazivanja realnih brojeva ona iznosi nakte preka deval citara. U Mnaram abilku broj zauzima 5 bajtova. Pošta ja u pitanju eksponencijalna notacija mantisa zauzima 4 bajta a eksponenat 1 bajt što je uobičajana tačnost za 8-bitna računare i programski jezik bajzik

(B Tomić)

Ko je uvredio IBM-a?

Javljam vam sa povodom napisa vase novinarke drugarice Jelene Rupnik pod naslovom „Vita jela zalen bor" računari br 6

Lapo je i za pohvalu jo što ima onih koji žele da budu drugačiji i da nake teme posmatraju na nov način. Problem je u tome što iakav jedan pristup irazi mnogo vise od puke želje — trazi da autor i sam

bude neka nova i svakako dobro verzirana ličnost. Čini mi se da primer koji nam pruža drugarica Rupnik ilustruje ko i kako to ne treba da radi. Koristeći izraze kao što su „šiznuti" i „šizneti" ona varovatno pokušava da bude neposredna i neopterećena konvencijama. Ni to nije bi samo po sebi bilo loša kada ne bi služilo pokriveru nedostatka i duha i nekog stvarnog poznavanja onog o čemu tako smelo i autoritativno piše. Zdravo narodni humor ipak nije zamena za otsustvo duha.

Postavka da narod kupuje SIN KLERA malo bolje storeci nared KOMODORA profesionalci BBC B a snobovi HEWLETT PACKARD MACINTOSH i IBM u najmanju ruku ilustruja ratku površnost autora. Ovako smatu tvrdnju bih potkrepio sledećim predlogom neka drugarica Rupnik sedne za narodni SIN KLER i neka na njemu obradi te tekstove po osam iti daset sati na dan mesec iti više dana uzastopce. Zatim neka isto ponovi sa računarom srednjih klasa pe onda sa profesionalnim BBC računarom. Na kraju kao varovatno ugrožanom pripadniku amaterskih masa i ukoliko je saviadata svoje gađenje prema toj ogavnoj mašini muškog roda ja je kao snob pozivam da sedne za gadni i klasno diferencirani IBM i ponovi prethodni test i to njoj za ljubav sa programom koji košta kao i 22 SINKLERA. Kada i ako nakon toga obradi rezultate bica joj nadam se jasnija da su snobovi doste često ljudi za koje taj računar predstavlja orude rada (Bez zlata) i da su iz tog razloga zainteresovani da svojih osam iti više sati rada provedu za takvom mašinom koja će ih u najvećoj mogućoj meri osloboditi napora zmišljanja o zlatu i koja će im podici produktivnost. Koliko je meni poznato upravo to je i bila osnovna svrha računara.

Iz tog ugla posmatrano nisu ti daleko veći snobovi upravo oni jadnici koji moraju da obijaju programe (koji su često na nivou dlota) u kojima je cilj da izbegnata zlog mrava uhvatite duha iti ne padnata u rupu i potrošite jedan od svoja tri života a preve žrtve oni ljudi koji bi želeli da od računara imaju i neka praktične koristi ali na mogu sebi da dozvole tako skupe mašine? Kratak pregled naših časopisa će vrlo brzo pokazati da se po mojoj slobodnoj proceni baram 60% prostora i barem 90% oglasnog prostora u njima posvećuje igrama a na izbližnim korisničkim programima Piratstvo je bez sumnje funkcija našeg relativnog siromaštva ali su zato predmati piratstva sigurno stvar našeg duhovnog siromaštva. Obijaču sa baskrejne i jednoličnе igre jer ogromna većina vlasnika računara ih i ima da bi sa sa njima igrale a na da bi sa njima ponešto pored igre i uradile — pa zar zato da ih žalimo? Pamet nije bi uradili da su desetinu tog novca utrošili na učtanjvanja u biblioteka koje zvrje prazna. Lično bih bio zadovoljan kada bih znao da se barem 10% računara kod nas u privatnim rukama koristi i za malo pamatnija stvari od turanja baba i žaba.

Svetsko tržište je odavno prihvatilo ponuđu koju sam uputio drugarici Rupnik koja pored ekonomske ipak sadrži i namensku komponentu. Pogledajte bilo koji katalog programa za SINKLER/KOMODOR računare pa ga zatim uporedite sa katalogom namenjenom recimo IBM računarima prvi će se sastojati 35% od igara i 5% od korisničkih programa dok će stvari u drugom stajati potpuno obrnuto. Pošto svetsko tržište na zavisi od lokalnih hirova time se jasno stavlja bo zhanje ko šta treba da kupi u funkciji ličnih POTREBA stvari koja je kod nas potpuno napoznat pojam. Za ovih par godina unazad samo sam nekoliko puta imao prilika da čujam konstataciju u smislu:

Kupicu računar GXL jer on odgovara mojim potrebama. Ko hoce da se igra i na treba da ima bilo šta više od SINKLERA iti KOMODORA mada sa ovaj drugi može i ta kako dobro upotrebiti i za ozbiljniji rad. Na žalost još nisam video oglas nekog pirata koji nudi racimo programa za obradu teksta na KOMODORU.

Voleo bih da mi drugarica Rupnik bostojne nastednica Minjana Bobić po pitanju računera odgovori za koji još operativni sistem na svetu postoji maker i deo korisničkih programa koji postoja za IBM ov MS DOS. Da li je moguca da su IBM PC i njegovi brojni klonovi proizvedani u preko četiri miliona primeraka samo za snobove po svetu iti ja pak subjaktivna i carinska stabost to što je on za nas jednostavno praskup i nedostižan?

A da li je zaista praskup? Kade se malo pre računamo uz pomoc cena u Engleskoj vidačamo da jedan KOMODOR sistem (računar dve disk jedinice monitor i štampač) košta približno 700 pora danja radi jedan ADVANCE 86 B sistam (kompatabitan sa IBM-om sa 128 umasto 64 kB RAM a i dalje proširiv na 768 kB dve diska od po 360 kB monitorom i štampačam EPSON 80 FX ta paketom programa) košta oko £ 1 500 dok slično konfigurisan IBM PC može da sa dobija ge oko £ 1 800 Narzgled, razlike su poveća ali samo naizgled. Razlika u ceni izmedu KOMODOR 64 sistama i ADVANCE 86 B nije 3 1 vec gotovo da je ni nama obzirom na paket programa koji biste za KOMODORA morali da do kupite da bista postigli jednakost — ukoliko takvih programa za KOMODOR uopšta i ima. Šteta je ali na žalost ipak istina da tzv sistemske potreba za ozbiljna korisnička programe dalako nadmašuju kapacitate SINKLERA i KOMODORA. Vecine osvedočeno dobrih programa (LOTUS 1—2—3 FORMULA 1 MULTIMATE razni kompajleri) zahtava najmanja 256 kB RAM a i neretko dve disk-jedinice od po 360 kB. Ako nekome LOTUS 1—2—3 može da obavi posao da li je onda takva osoba snob ako kupi računar na kome može upotrabiti taj program? Ili je možda nezgoda u tome što oni koji donosa zakone koji onemogucavaju kupovinu ozbiljnih računara (a SINKLER i KOMODOR to nisu jer kao takvi nisu nit bili zamišljeni niti proizvedant niti se kao takvi prodaju) nisu najbolje upoznati sa onim u vezi čega donosa zakone? Kako može naša omladina da nauči COBOL FORTRAN LISP i druge jezika bolje od BASIC-a kade na postoji tehnički osnov za to?

I kako tako površne i paušalne da ne kažem neozbiljne postavke uopšte mogu da nađu mesta u časopisima kao što je Računar koji je na sebe preuzeo ozbiljne nužne i veoma odgovorne zadatke? Zašto se nekima dozvoljava da vežbaju kucanje na skupoj, naivnoj, hartiji? Umesto da se isti taj prostor posveti takstovima koji upućuju na ozbiljniji rad kao recimo onoj koji su napisali drugovi Spasić Perzić i Ristanovic

S drugarskim pozdravima
D V računarski snob
Br lična karta 34182

P S U krugu snobova u kome se takode razmanjuju piratrani programima što i nije čudno obzirom na prosačnu cenu jednog takvog programa koja iznosi kotiko i BBC B računar nezamistivo je naplaćivanja bilo kakve usluge kopiranja a normalna je stvar da sa krug odmah obavesti o pripaču nekog novog programa. Neosporno to je i dalje piratstvo ali na nakom drugam osnovu — damokratija snobova?

Poštovani V.D. računarskog snoba interesantno je da je Vaša pisma ska 4,5 puta duže nego teket na koji sa žalite i u kojam se izgred budi rečeno nigda na kaža da SVI vlasnici IBM PC mareju biti snobovi. Stoga sa umaljevata da se u najskarije vreme javite drugarici Rupnik radi izdavanja potvrda da ne pripadate računarskoj snobovštini. P S Niste dobre obavešteni: ciji igra nija da izbegnata zlog mrava, mrav je dobar a škorpija su ljuta

I car i kralj i cela monarhija

Pišem povodom članka Amstrad protiv Komodora. Odmah u uvodu sa vidi greška! Stevio je nedavno pod istu lupu amstrad i nekrunisanog kralja medu kućnim računarima – komodora 64.

Mislim da je nakrunisanog kralja (komodora 64) trebalo krunisati kao cara!! On je nastao 1981 a amstrad čak četiri godina kasnije! Stoga mislim da nije trabalo uporedivati ova dva mikroračunara jer se razlikuju po godini proizvodnje i mikroprocesoru (amstrad koristi Z—80).

Kakav bi test ispao da ste poredili BBC B i etari 800XL? Zašto niste uporadivali amstrad i orika??

Mislim da ste bacili šagu na komodor 64 bez prava. Kako sta ga samo hvalili u Računarima 2 sa. Nikog ne ostavlje ravnodušnim. U vrame kad sa pojavio pošto je pretekao sve dotedašnje računara on je POSTAO i OSTAO naj naj mikrac!

Mislim da ovakvi testovi više ne bi smali da vam izlata jer možata pokolabati i najčvršceg kupca.

Ako vec uporedujete računara različitih procesora tada uporedite amstrada i komodor 128 (ako ga nabavite).

Vidoja Veselinovic

Aca nije fer

Javljam se prvi put i to u vezi programa Velika akcija. U prošlom broju Računara Čele Zoltan je imao primedbu da ja program Velika akcija već bio distribuiran. Vi ste mu ne to odgovorili da su za to krivi pirati. Međutim nista u pravu! Alaksander Radovanovic ima svoju programarsku (i piratsku) firmu Computerland. Naručivši katelog od A. Radovanovica video sam da je i on sam prodavao Vaš ku akciju.

Mislim da je Velika akcija nezasluženo osvojila drugo mesto. Uzgred vreme bih se zahvalio drugu Radovanoviću ne programu Komprasija (Računari 6) koji sam ukucao u kompjuter i savršeno mi funkcionišе. Pomoću Komprasija sam napravio avanturu. Dvorac koja ima mnogo slika i većim delom je pisana u mašinci. Smatram da je bolja od Velika akcija ali je na žalim nikom pokazivati jer je čuvam za drugi konkurs. Molio bih da mi odgovorite na pitanja koja ne zanima samo mana nago i većinu hakara širom zemlja. KADA CE SE ODRŽATI DRUGI KONKURS ZA NAJBOLJEG YU PROGRAMERA I KOLIKO PROGRAMA MOŽEMO POSLATI NA KONKURS?

Nasic Aleksandar

„Velika akcija" i gresi mladosti

Tačno ja da sam pripadao piratskoj firmi Computerland. Međutim maja 1984 godine firmu preuzima drugi čovek. Program Velika akcija pisan je u toku leta iste godine dakle posle moje piratska avantura.

Činjenica ja da je program raden na Computerland-ovom računaru jer ja nisam posedovao svoj. Radna verzija programa ja ostela njima, međutim na moj zahtev nisu prodeli ni jeden jedini snimak. Program je procurio zahvaljujući jednom prijatelju koma je dat radi ocena i sugestija. Pravu konačnu verziju programa koji je učestvovao na konkursu i koji ja prikazan na dodali nagrada na poseduja niko od preprodavaca. Verzija koja kruži medu korisnicima ja nedovršena. Moguće je preći 60% igra postoja neki bagovi a i naslovni ekran predstavlje radnu verziju. Sve to znaju oni koji imaju taj program.

De nisam imeo nikakve zle namera potvrduje i činjenica da sam Veliku akciju pisao sa namerama da sa emituje u radio amiši. Ventilator 202 u septembru 1984. Program nije emitovan jer me ja Zoran Modli smatrajući da je šteta da toliki trud ode samo u eter nagovorio da ga pošaljem na konkurs. Posta otpritka mesec dana došlo je do jedne smašne zgode. Pojavila se

➔

grupa pirata sa svojim potpisima na Velikoj akciji sa željom da je emitujú u istoj emisiji. Čakla

a) Od maja 1984 na pripadam Computerlandu niti su oni proda vali moj program mada ga imaju u katalogu

b) Na programu nisam ostvario nikakvu zaradu

c) Spectrum od sada koristim isključivo za igru

Aleksandar Radovanović

Jelena suzama ne veruje

Možda je postalo mrtvo na tekstovi vaša saradnica J Rupnik trifraju samo muški deo gradan stva ali ovo je pravi štvo svaku ma ru Mislim konkretno na njen tekst

Obračun kod Tjunngove mašina U tom tekstu ona verovatno poku šava da kaže nešto ali šta to mi još uvek nije jasno Ona javno vređa kompjuterske stručnjake koji se po njoj pojave na TV da oskvrnjuju o kućnim računarima Čak i relativno neupućeni ljubitelji mogu da pri mete da ovo pojma nemaju o čemu govora Propustili su da se pri hodno kod lokalnih hakera obave ste o čemu se radi (????!!)

Dobro mohm vas na šta ovo liči? Ako smo dotle došli da doktori i magisti nauka treba da uče od hakera onda kuda ida ovo dru štvo? Ima još! Na primer Oni koji sa profesionalno bave računarima na vole da im sa po dvorišu muva ju tamo neki hakeri Oni strepe da se na kraju ne isposlavi da je njihovo čedo-dinosaurus u stvari malounno

Iz ovog citata se može videti totalna neupućenost drugarice J R u tu problematiku Ovom naro čito poslednjem računarcem vero vatno je htela da kaže da veliki sistemi ne mogu biti bolji od mikro računara Drugarice Rupnik, hteo bih da vas pitam mogu li uopšte da se poredi IBM—370 koji ima pet naestak terminala sa jednim kuć nim mikroračunarom pa makar on bio i vrhunski te vrste npr BBC ili QL

Drugarica Jelena mislim da bi vam stvarno bilo bolje da podete na mah kurs iz osnova računarske teh nike Ako ne mislite da učinite tako nešto zašto bolje ne počnete da pišete o onoma šta vrlo dobro znate — a to su igra kuvanja ručka i slično — ili da sa konsultujete sa nakim od kolega saradnika iz ra dakcija pre nego što počnata da pišete stručan tekst

Uzgred mogli bista angažovati i Ljubu Moljca da vam učini tekstove malo smešnijim!

Račić Samir
Dragošava 3 VI
18300 Pirot

Žensko koje piše

Tačno sam znala da će m počnete da obaviljujete pisma jedna od žrtvi napada bih i Jelena Rup nik Od kad su počeli da se poja vljuju njeni tekstovi u Računari ma znala sam da piša drugačije izazovnija i svežija nego što bi smela Meni se to jako svida ali sam bila sigurna da će biti čitava gomila ušlogrejenika i kojekakvih ti pova koji neće moći da podnesu njene tekstove da na pominjem zadrte muškarčina koje sigurno in tira njen neolemnisički jon

Kako ste vi kao redakcija uopšte mogli da objavite ono pismo Saše Kovačevića? Ja sam to shvatila kao da vi sprvujete Jelenu Rupnik u odnosu sa njenog učešća u vašem radu To je apsolutno izdajnički Gotovo sam sigurna da u redakciji sede sve sami muški sovinisti i računarski čistunci koji su jedva dočekali pismo kakvo je Saša Ko vačević napisao da bi sa odmah ogradili od onog što ih ugrožava a to je žensko što piše Niste mogli na bolji način da potvrdite Jelenine taze iz članka Računari su mu škog roda

Nadam se da moja pismo neće biti shvatih kao histeričan napad na sve muškarce i da ćete ga objaviti da bi se videlo da ima i nas kojima se Jelenini članci veoma sviđaju

Ana Trojanović
Otona Župančića 1
Novi Beograd

Sledžija iz javnih glasila

Priličo sam se obradovao kad sam video da ste uvah pisma čitala ca pardon load pismo u Računa re Baš taj prvi broj u kome je ta rubrika objavljena ma je i ponukao da napišem ovo pismo

Pročitao sam pismo koje je na pisao Saša D Kovačavic a koje sa tiče pisanja Jelene Rupnik Na stra nu to kako Jelana piše Ah ako sa dobro secam to je onaj isti Saša koji vec duže vremena koristi svaku priliku da opali po Dubravki Marko vic preko lalasa Indexa 202 čiji je najveći grih što se zove Duca a bio je izuzetno akhvan i u imtiranju i spominjanju Sonja Savić Ako se dobro secam to je lakoća onaj isti Saša koji prilično često gnjavi jed ne reportere Indexa E pa mislim da uopšte nije slučajno što su i Jelana i Dubravka i Sonja i rapor tarke ženskog roda

Pretpostavljam da je doste viso ko obrazovanja uskratilo Saši pnit ku da davojka prepada i plaši po hlovima pa njegova podsvesti to kanališe prema verbalnim i pisanim napadima po raznim sredstvima in formisanja a sad i u vašem časopi su Zar Saša zaista misli da se to tako radi? I zar na bi bilo jedno stavnije da Saša sazna telefona pomenutih djevojaka i da svoja obrušavanja obavlja u miru privat na veze?

Jalačić Boško
Konjarnik Beograd



Ponovo monitori

Tekst o monitorima objavljen u prošlom broju računara izgleda da je raspalio strasti u mnogo većoj meri nego što je to redakcija pre dvideli Jevlo nam se nekoliko dasalima vlasnika spakirama i komodora koji traže precizan od govor na pitanja koji monitor odgovara njihovoj mašini i kako ga spojiti

Da krenemo od spekiruma zelenog monitora Problem je na ravno u tome što spectrum na ma nikakvog monitorskog izlaza Ipak uz malo poznavanja elektroni ke moguće je skoro bez ikakvih probleme povezati spekirum i zeleni monitor Najčešće je dovoljno uzeti signal ispred FR modulatora Što se tiče kolor monitora proce dura je potpuno ista s tim što treba strogo paziti da monitor ima kom pozitni ulaz — nikako ne RGB

Se komodorom ima znatno manje probleme To jest jedno mogu da smatraju različiti konekto ri Zeleni monitor može bilo koji a kolor mora obavezno da bude kompozitni RGB baš kao i kod spekiruma neće valjati Ipak nije sve tako roze ni kod komodora Dobicete divnu sliku ali bez tona Kod zelenog monitora ti nema le ka dok kod kolor monitora možete da birate ili ćete kupiti originalni komodor po ceni od oko 210 funti ili ćete naći neki kompozitni koji ima i zvuk Jedan od takvih je FIDELITY CM 14 12 MHz RGB-/COMP/SOUND sa cenom od 190 funti

Važno je primetiti da su spak irum i komodor računari se relativno slabom rezolucijom pa će svaki najfiniji monitor odlično odgovarati Cena zelenih monitora se obično kreću između 70 i 150 funti Da pomenamo neka PHILIPS 12 GREEN (90 funti) SANYO SM12N GREEN 15MHz (90 funti) SANYO DM8112 GREEN 18 MHz (130 funti) KAGA 126 (120 funti) Ako vam sa više svida amber (žuti) monitor njihova cena su obično nešto malo veća Racimo NOVEX AMBER (125 funti) ili KAGA 12A (140 funti) Kolor monitora au na ravno mnogo skuplji Za model niža klasa trabe izdvojiti između 200 i 400 funti Pored već spomenutog originalnog komodorovog monitora i FIDELITY CM 14 pomenimo još MICRIVITEC 1431 MZ spe cijalno namenjen spektrumu (230 funti)

Nemaskirani interapt

Zoran Jovanović iz Novog Mesta se interesuje za nemaskirani interapt procesore Z80 i za grešku koja se nalazi u spektrumovoj rutini za tretiranje nemaskiranog interapta

Kada procesor Z80 preko nožica NMI primi zahtev za nemaskirani interapt on prekida svoj redovan posao stavlja na stek sadržaj PC (PROGRAM COUNTER) registra i skače na adresu 66 heksadekadno U spektrumovom ROM u na toj adresi se nalazi sledeći program

```
RESET     PUSH AF
          PUSH HL
          LD HL (23728)
          LD A H
          OR L
          JR NZ NO RESET
          JP (HL)
NO RESET  POP HL
          POP AF
          RETN
```

Program ima jedan vrlo glup beg umesto JR NZ trabalo bi da stoji JR Z Konstruktori su bili pre dvideli sledeće uzme se sadržaj sistemske promenjive NMIADD (23726 23729) Ako je taj sadržaj jednak nuli ne radi se ništa Pravilno je da se na NO RESET a odatle se nastavlja izvršavanje redovnog programa U suprotnom ako nije nula skače se na adresu na koju ukazuje NMIADD

Korist od nemaskiranog inte rapta može da bude u sledećem spektrumu dodamo jedan specijalan taster za RESET spojen jed nim krajem na NMI nožicu procesora a drugim na masu u RAM stavimo jednu kratku rutinu za nor malizaciju sistema a u sistem ve riablu NMADD adresu te rutine posle toga možemo recimo da tastaturno neki mašinski program Ako stvari kratu naopako pritisne mo reset taster izvrši se rutina za normalizaciju sistema i na kraju operativni sistem preduzima kontrolu Naravno sve što je bilo u memoriji ostaje netaknuto U toma je i sva prednost ovakvog rešenja u odnosu na onaj kada pomoću jed nog testera spojimo RESET nožicu sa masom

Sva bi to lepo radilo da nema onog glupog bage Da bi sa to ispravilo nama druga nago da ceo sadržaj ROM-a (sa ispravljanom greškom) prekopiramo u jedan 27128 aprom stavimo na mesto ROM -a Te idaje je izuzetno dobra po red ostalog i zbog toga što sa u ROM u nalazi oko kilobajt praznog prostora Tu možemo sasvim lepo da stavimo celu rutinu za normali zaciju sistema pa čak i još nekoli ko korisnih programa Rutinu za normalizaciju sistema možete naći u RAČUNARIMA 1 pod naslovom Ekskluzivni reset U sledećem broju Računara Petar Putnik ja dan od pobednika Galaksijinog konkursa za najbolje YU programa ra pisaća o mogućim preprav kama spektrumovog ROM a

ZABAVA★UČENJE★POTREBA

Iz serije 700, predstavlja vam e novom, nižom cenom, kompaktni personal kompjuter SHARP MZ — 731.

Kesetofon — Cetvorobojni štampeć (crna, pleva, crvena, zelena) — Kepecitet memorije 64 KB RAM.

Uz MZ 731 možete posebno neručiti:

„Zeleni" monitor po ceni od 420 DM ili TV koior monitor po ceni od 840 DM.

Dinerske dažbine cca 65%.



Mercator — Mednarodna trgovina

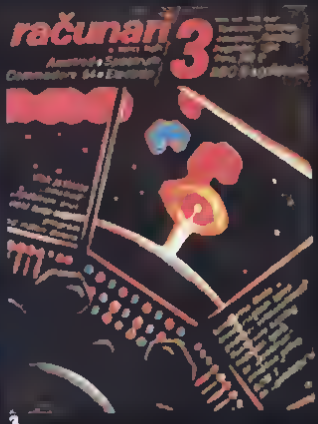
Ljubljana
Telefon 061/328 441 lokal 839



1



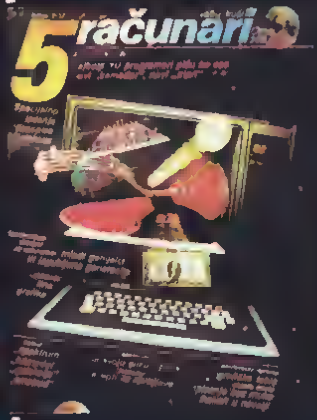
2



3



4



5



6



7

NARUDŽBENICA

GALAKSIJA Bulevar vojvode Mišića 17 11000 Beograd

Ovim neopozivo naručujem sledeća specijalna izdanja časopisa „Galaksija" 1 — 2 — 3 — 4 — 5 — 6 — 7 (zaokružiti odgovarajući broj), po ceni od 200 dinara po primerku. Iznos od ukupno _____ dinara uplatiću poštaru prilikom preuzimanja pošiljke — POUZEĆEM

Ime i prezime _____

Ulica i broj _____

Broj pošte i mesto _____

Datum _____ Potpis

1 2 3 4 5 6